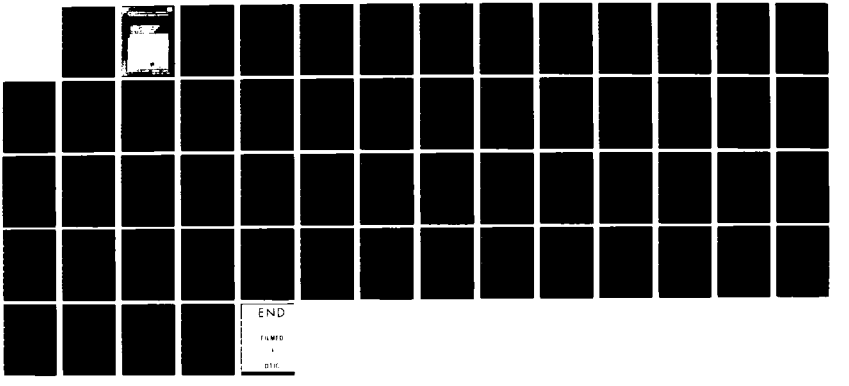
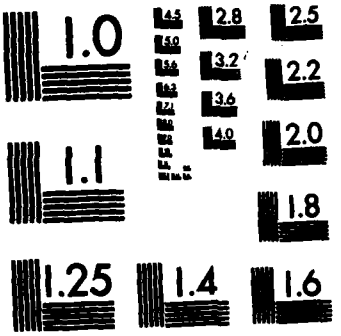
AD-A124 045 A SIGNAL AVERAGER INTERFACE BETWEEN A BIOMATION 6500 TRANSIENT RECORDER A.. (U) ILLINOIS UNIV AT URBANA COORDINATED SCIENCE LAB L T SPECHT JUN 80 R-885 1/1

UNCLASSIFIED N00014-79-C-0424 F/G 9/2 NL





UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A124 043	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* A SIGNAL AVERAGER INTERFACE BETWEEN A BIOMATION 6500 TRANSIENT RECORDER AND A LSI-11 MICROCOMPUTER		5. TYPE OF REPORT & PERIOD COVERED Technical Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) L. T. Specht		8. CONTRACT OR GRANT NUMBER(s) N00014-79-C-0424
9. PERFORMING ORGANIZATION NAME AND ADDRESS Coordinated Science Laboratory University of Illinois at Urbana-Champaign Urbana, IL 61801		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Joint Services Electronics Program		12. REPORT DATE June 1980
		13. NUMBER OF PAGES 55
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Signal averager
Transient recorder
Computer Interfacing

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This report describes the design and implementation of a versatile and compact signal averager interface between a Biomation 6500 transient recorder and a LSI-11 microcomputer. The design allows for fast signal averaging in excess of 1 kHz and is also easily software configurable and controllable. The interface has been incorporated into an LSI-11 system using the RT-11 operating system and found to perform very satisfactorily.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

A SIGNAL AVERAGER INTERFACE BETWEEN A BIOMATION 6500 TRANSIENT RECORDER AND A LSI-11 MICROCOMPUTER

by

L. T. Specht

This work was supported in part by the Joint Services Electronics Program (U. S. Army, U. S. Navy and U. S. Air Force) under Contract N00014-79-C-0424.

A SIGNAL AVERAGER INTERFACE BETWEEN A BIOMATION 6500 TRANSIENT RECORDER AND A LSI-11 MICROCOMPUTER

L. T. Specht

Department of Electrical Engineering and
Coordinated Science Laboratory

Abstract

This report describes the design and implementation of a versatile and compact signal averager interface between a Biomation 6500 transient recorder and a LSI-11 microcomputer. The design allows for fast signal averaging in excess of 1 kHz and is also easily software configurable and controllable. The interface has been incorporated into an LSI-11 system using the RT-11 V3B operating system and found to perform very satisfactorily.

1. INTRODUCTION

This is the description of a versatile signal averaging system based around a Digital Equipment Corporation (DEC) LSI-11 microcomputer and a Biomation 6500 transient recorder (TR). With the widespread utilization of LSI-11's in various experimental configurations, it is convenient to have a signal averager (SA) plug-in that would interface a transient recorder front end unit with the computing power of an LSI-11. This would eliminate, in many cases, the need for an external signal averager mainframe in order to incorporate a TR into an existing LSI-11 based system. Besides providing a rather large cost savings, this results in a more compact system that is both versatile and easy to use. The SA interface described herein is compatible with either the DEC LSI-11, LSI-11/2 or LSI-11/23 CPU and any system based on one of these CPU's and which will henceforth be referred to as simply LSI-11.

At this point a brief description of what is meant by a signal averager and transient recorder may be in order. Basically, a signal averager adds together waveforms that are synchronized in time so that any coherent features that are present will add constructively; whereas, any incoherent features will add destructively. In this way the ratio of signal to noise (SNR), or coherent to incoherent features, can be increased from that which it is for any given waveform. Typically the

addition is done digitally which necessitates the need to be able to digitally represent the original analog signal. This is done by digitizing the analog signal at N time intervals of Δt each with an analog-to-digital converter (ADC) so that the total time sampled is $N\Delta t$. Thus for each sampled point the time resolution is defined by Δt and the amplitude resolution is determined from the number of bits in the ADC. For fast signals the sampling interval Δt is usually too short for any type of real time processing so that some form of buffer memory is used to temporarily store the digitized waveform before it can be processed. This combination of ADC, sampling clock, and buffer memory is referred to as a transient recorder. For the Biomation 6500 transient recorder the main specifications are: (1) minimum sampling interval of 2 ns, (2) total of 1024 sampled points, (3) ADC resolution of 6 bits, and (4) maximum output rate for the stored waveform (1024 words of 6 bits each) of 500 ns/word. As an aside, one should note that the above restriction of non real time signal processing of fast signals can be eliminated by incorporating a transient recorder and signal averager into one integral unit.[1] The main impetus for real time processing is to obtain the maximum repetition rate possible since the SNR improves as the square root of the number of waveforms averaged.

In typical experimental configurations some form of additional data manipulation is usually required after the averaged

waveform has been obtained. This may include addition and subtraction of waveforms to remove undesirable coherent features, integration, differentiation, Fourier transform, graphical display, handcopy output, waveform storage, etc. When used in conjunction with an LSI-11 system all of these features and more are readily programmable and changeable as the user's needs change. Also since the SA interface is software controllable it is readily amenable to incorporation into an automated environment.

In view of the above considerations the main features that the SA interface incorporates are as follows.

1) Plug-in capability - everything is contained on a single quad size printed circuit board.

2) High data taking rate - this allows for a 1-1.5 kHz repetition rate for transferring a 1024 point acquired waveform.

3) Low CPU dead time - at the maximum repetition rate there is only approximately 2% dead time for any length of time.

4) No premanipulation of stored data - the signal averaged waveform is stored in either single precision (16 bit) or double precision (32 bit) integer format.

5) x-y analog outputs - two 10 bit DAC CRT monitor outputs are provided for either real time viewing of the signal averaged waveform or programmed display of the memory contents.

6) Software controllable and configurable - see the software section for the various options and configurations that may be implemented by the CSR.

7) Ease of operation and extensive data manipulation - this is basically a characteristic of the LSI-11 system and corresponding software.

As a final note, while the SA interface described here is specifically for use with the 6500 any other TR that uses a similar means of data transmission can be utilized as a front end unit. Namely, the 6500 uses a word-serial, bit-parallel data transfer under a "handshake" control. See the circuit description for more information if the SA interface is to be used with another TR.

2. HARDWARE OVERVIEW

The basic block diagram of the SA interface is illustrated in Figures 1-5. There are five major functional sections: 1) arithmetic logic unit (ALU), 2) memory, 3) x-y DAC, 4) DMA/ interrupt controller, and 5) master controller. Each of these five sections will now be briefly described, a more complete discussion can be found in the circuit description section.

The ALU section is basically a 16-bit adder whose B input is the memory data bus (MEM Ø-15) and whose A input can be either the 6500 data bus (PERDATA Ø-5) or the SA data bus (DATA Ø-15). Both the A and B inputs can be set to zero via the data output register (DATOR) and the sum latch, respectively. The adder output is written into the sum latch which can be placed back on the memory data bus, and the data input register (DATIR) which can be placed back on the SA data bus. This allows the ALU to operate on the 6500 data bus or the SA data bus in one of three ways: 1) add to memory, 2) overwrite memory, or 3) write zero to memory.

The memory consists of four 1k×4 bit static RAMs arranged as 1k×16 bit words. The memory address (MA) is generated by the memory address counter and stored in the memory address register. For display purposes the MA counter may also be continuously clocked at a 1 MHz rate.

A x-y CRT monitor output is available through two 10-bit DACs. The x-DAC monitors the memory address bus; whereas, the y-DAC monitors the 9 least significant bits and the most significant sign bit of the memory data bus, assuming that data is stored as signed z's complement fixed point numbers. Each channel provides a ±5V full scale output. A 12V inverter is provided for the DACs so that the only backplane voltages required are +5V and +12V. A z-blank output is also available. The x-y output can be used to monitor the memory contents either real time during signal averaging, or under program control.

The DMA/interrupt control section is capable of performing DATI, DATIB, DATO, and DATIO bus cycles which are input, output, and input/output (read-modify-write) transfers between the LSI-11 and the SA interface. This section contains the LSI-11 bus transceivers and logic to implement interrupt requests, address control, protocol, and DMA requests. Also contained in this section are five registers: 1) word count register (WCR), 2) bus address register (BAR), 3) control status register (CSR), 4) data input register (DATIR), and 5) data output register (DATOR). The device address and interrupt vector are switch selectable as illustrated in Figure 6 thus establishing the five register addresses and the two interrupt vectors.

A brief functional description of each of the five registers is given below.

1) The WCR is used to control the number of words transferred during a DMA cycle. It is loaded with the 2's complement of the number of words to be transferred and is incremented after each word is transferred. The DMA cycle is terminated when the word count reaches zero.

2) The BAR provides the memory address to or from which data is to be transferred. It is loaded with the starting address of the transfer and is incremented after each word is transferred.

3) The DATIR and DATOR are write only and read only registers sharing the same register address. These registers, as the others, can be accessed under program control but are typically used during a DMA cycle to buffer data into or out of the SA interface.

4) The CSR is used to control the functions and monitor the status of the SA interface. The SA interface may be configured in many different ways through the use of the CSR; refer to the software section for a description of each of the CSR bit functions.

The master control section provides the control structure for interconnecting the other sections of the SA interface. This includes handling communication with the 6500, keeping track of the number of processed waveforms, initiating DMA requests, and implementing the various configurations setup through the CSR.

Typical cycle times are 700-900 ns per word, which for the 1024 word waveform stored in the 6500 allows a maximum repetition rate of 1-1.5 kHz. At this rate a SNR improvement of 100 can be obtained in 10 seconds. Also at the maximum rate a DMA cycle is requested every .5 seconds which takes approximately 8 ms for a double precision transfer thus giving rise to approximately 2% CPU dead time.

3. SOFTWARE

This section discusses the SA interface from a software point of view with the intention that the reader may learn how to use the SA interface without having to become familiar with all of the hardware. It is assumed that the reader is somewhat familiar with the LSI-11 software, if not, one can consult the "LSI-11 Microcomputer Handbook".[2]

The SA interface is programmed through the five registers contained in the DMA/interrupt section: WCR, BAR, CSR, DATIR, and DATOR. Each of these registers has a unique address established by the device address, and as such each is accessible under program control. The WCR and BAR are used during DMA cycles to keep tract of the number of words transferred and to provide a memory address to or from which each word is transferred. The WCR is loaded under program control with the 2's complement of the number of words to be transferred and the BAR is loaded with the starting address. The DATIR and DATOR can also be accessed under program control to read or write information out of or into the SA memory. Each time either the DATIR or DATOR is accessed the memory address is advanced to the next location. Normally the DATOR and DATIR are used during DMA cycles to buffer data into or out of the SA interface. The CSR is used to control and monitor the status of the SA interface. A brief description of each of the 16 CSR

bit functions is given in Table 1. Each bit can be read or written under program control. Bits 4 and 5 of the CSR are special in that they are the extended memory address bits XAD 16 and 17 making the SA interface compatible with the memory management unit (MMU) of the LSI-11/23.

Two interrupt vectors are also selectable on the SA interface. They are typically used by interrupt requests generated at the end of a DMA cycle or due to an error condition to point to interrupt service routines. A very simple program illustrating the use of these registers and the interrupt vectors in normal operation is provided in Figure 7.

Data can be transferred during a DATIO cycle in either single or double precision format. Single precision format results in a 16 bit signed 2's complement fixed point integer (INTEGER*2); whereas, double precision format results in a 32 bit result (INTEGER*4). Since the data is placed in standard integer format no premanipulation of the data is necessary. The double precision format also allows very extensive signal averaging for long lengths of time without loss of precision.

Besides the normal DATO or DATIO cycle transfers, data may also be transferred into the SA interface during a DATI or DATIB cycle. This allows the SA memory to be used as a 1k scratchpad area with the capability of either simple read-write operation or single step add to memory operation. It also allows loading the SA memory with information to be displayed using the x-y monitor.

4. CIRCUIT DESCRIPTION

This section provides a detailed description of each section of the SA interface. While reading it may be useful to look at the appropriate schematic indicated for each sub-section. In this description the convention of placing an H or L after a signal mnemonic indicates either a high or low active signal.

4.1 Arithmetic Logic Unit (ALU) - Figures 8 and 9

The ALU section uses either the SA data bus (DATA Ø-16) or the 6500 data bus (PERDATA Ø-5) to perform one of three functions on the contents of the SA memory: 1) add to memory, 2) overwrite memory, or 3) write zero to memory. The 6500 data bus is terminated with 150 ohms and goes to one input of a 16 bit 2-to-1 multiplexer (MPX) consisting of four 74LS244 non-inverting octal buffers and 3-state line drivers connected in series parallel. The 74LS244 is used because of its 400 mV noise margin. The other MPX input is the SA data bus which is selected by SACYC H. The high byte and low byte of the MPX are latched by DSTBHB H and DSTBLB H respectively into the data output register (DATOR-74LS273). The output of the DATOR (ADDA Ø-15) which may also be set to zero by asserting DZERO L, goes to the A input of a 16 bit adder (74LS283). The adder's B input is the SA memory data (MD) bus (MEM Ø-15) which monitors the contents of the SA memory. The adder output is

latched into the data input register (DATIR-74LS374) by DATLD H and into the sum latch (74S412) by SUMLD L. The output of the DATIR goes onto the SA data bus when DATEN L is asserted and the output of the sum latch goes onto the SA MD bus when SUMEN L is asserted. The SA MD bus is active pull-down and may be set to zero by asserting SUMCLR L on the sum latch. The carry-in and carry-out of the adder are CIN and COUT respectively, these are used for double work transfers during a DMA operation.

4.2 Memory - Figure 10

This section is based around a 1k word by 16 bit memory array composed of four 2114 (MCM2114L) 1k×4 bit statis RAMs with typical access times from 200-450 ns. The memory input and output is via the 3-state SA MD bus with the memory being enabled by CS L. Information is written into memory by application of WP L. The memory address (MA) is provided by the MA register (74LS374) which is loaded by MACLK L, which also increments the MA counter (74LS393) to the next MA location. When the MA counter reaches 1024 it generates 100 ns pulses NWOVF H and RESET L which reset the MA counter to zero. The SA memory may also be reduced from 1024 words to 512, 256, or 128 words by setting the memory size switch (S1) to the appropriate position. For displaying purposes the MA counter may also be continuously cycled at a 1 MHz rate by application of MEMDSPLY H (see DAC section).

4.3 Digital to Analog Converter (DAC) - Figures 10 and 11

Two analog outputs, XOUT and YOUT, are provided for monitoring the contents of the SA memory on a x-y oscilloscope. XOUT is produced by applying the MA lines to the input of a 10-bit DAC (AD561) which in turn drives a fast settling op amp (AD509). This provides a full scale output of -5.00 V to +4.99 V for MA values of 0 through 1023 respectively. YOUT is produced by applying the 9 LSBs and the inverted MSB (sign bit) of the SA MD bus to the input of another 10-bit DAC and op amp combination. This provides a full scale output of -5.00 V to +4.99 V for SA MD bus values of -512 to 511 respectively. This output does not correct for rollover caused by the SA MD bus exceeding these values.

A 12V inverter is provided on board so that +12V and +5V are the only backplane voltage needed. The inverter uses a 80 kHz charge pump oscillator built around a LM311 to pump a 1 mH inductor. This is filtered to provide -15V which is then regulated to -12V with a 7912 regulator. This voltage is used by both the AD561 and AD509 to provide XOUT and YOUT.

A TTL compatible z-blank is also provided which can be used to blank the oscilloscope trace except during a SA cycle or when MEMDSPLY H is asserted. A SA cycle is determined by SACYC H being asserted, and MEMDSPLY H is used to provide a flicker-free display of the uncharging SA memory contents.

4.4 Direct Memory Access (DMA) and Program Control - Figures 12-17

Since the implementation of this section is heavily based on the use of Digital's custom ICs (CHIPKIT), it is suggested that the reader unfamiliar with these ICs consult the "CHIPKIT Users Manual"[3] for further information and circuit descriptions. It is also assumed that the reader is familiar with the LSI-11 microcomputer, and if not, to consult the "LSI-11 Microcomputer Handbook".[2] The first part of this section will deal with those components that are common to both the DMA and program control data transfer, whereas the latter part will deal with those that are unique to the DMA transfer.

Four DC005 transceivers are used to interface the 16 BDAL lines to the 16 SA data lines. Their receive or transmit status is determined by REC H or XMIT H being asserted respectively, with receive being the normally active state. The transceivers also provide for device address and interrupt vector inputs determined by user selectable switches A12-A3 and V8-V3. These user configurable switch settings are illustrated in Figure 6.

When the proper device address has been decoded by the DC005, the protocal logic (DC004) is enabled to decode the proper bus synchronizing signals. SA data lines 1 and 2 are decoded to produce SELØ L - SEL4 L which select one of four SA registers. The direction of transfer into or out of the

selected register is determined by INWD L, OUTHB L and OUTLB L which are generated by the control lines BWTBT L, RSYNC L, RDOUT L and RDIN L; the latter three control lines are the buffered bus signals BSYNC L, BDOUT L and BDIN L.

Two interrupt channels, A and B, are provided by DC003 with channel A having a higher priority than channel B. The respective channel is enabled by asserting SA data lines 14 and 12 and then toggling CSRWHB H. The status of either channel can be monitored by ENAST H and ENBST H which form part of the control status register (CSR). After being enabled the appropriate channel interrupt request may be made by asserting RQSTA H or RQSTB H which in turn asserts BIRQ L. The daisy-chain bus signals BIAKI L and BIAKO L are then used to determine the priority of the interrupting device. After the priority has been determined the device vector is then placed on the BDAL lines. Interrupt requests are provided for completion of a DMA request, a bus timeout due to a non-existant address, and a user initiated request for device attention. The DC003 also buffers BINIT L to provide a SA initialization signal INIT L. Setting the CSR reset bit also generates a 1 microsecond INIT L pulse.

The bus timeout interrupt is provided so that the bus will not hang up for more than 10 microseconds if a non-existant address is placed on the bus. When the address is placed on

the bus a 10 microsecond one-shot is clocked by ADREN H. The one-shot is then cleared when RPLY H goes high to signify that the address has been accepted. If RPLY H does not go high the one-shot clocks the A interrupt flip-flop which produces (TOS + INIT) H and negates REQ H to release the bus. The A interrupt flip-flop also sets RQSTA H to initiate an interrupt request if the A interrupt channel has been enabled. RQSTA H may also be set directly through the CSR or through the attention bit of the CSR. Setting the attention bit asserts ATTN L which generates RQSTA H immediately if the device is inactive or at the conclusion of the current DMA or SA cycles.

The logic necessary for DMA bus arbitration is contained in the DC010. A DMA request is made by asserting REQ H which generates BDMR L. After the bus acknowledges the request the daisy-chain signals BDMGI L and BDMGO L are used to determine the priority of the requesting device at which time the device becomes bus master. The control lines DATIN L and DATIO L are used to determine whether a DATI, DATO, or DATIO transfer is to take place.

For a DATI transfer DATIN L is set low and DATIO L is set high; for a DATO transfer DATIN L is set high and DATIO is set high. For a DATIO transfer DATIN L can be either high or low and DATIO L is set low. When DATIO is asserted the falling edge of DATN H generates a 70 ns pulse, DATIOT L, to the DC010 to complete the output portion of the DATIO transfer.

After every fourth word is transferred the bus is released for ∿ 1.2 microseconds to allow other DMA devices to access the bus before bus mastership is regained and the transfer continues. This feature may be defeated by connecting CNT4 to +5V so that the device will transfer data in a continuous burst node. Note that if this is done the memory refresh must be taken into account if volatile memory is being used. All timing for the DC010 is provided by an 8 MHz oscillator (74LS132).

Once the DC010 gains control of the bus it asserts ADREN H to the bus address register (BAR) to place the memory address of the data transfer on the SA data lines. The BAR is loaded under program control with the starting address in memory of the data transfer and is incremented by two on each transition of ADREN H. The word count register (WCR) is loaded with the two's complement of the number of words to be transferred and is incremented on each transition of ADREN H also. When the WCR overflows it generates WCNTO H which negates REQ H and terminates the DMA transaction. WCNTO H also sets the B interrupt flip-flop which asserts RQSTB H so that if the B interrupt channel has been enabled initiates an interrupt request. Both the WCR and the BAR are contained in two DC006s cascaded to form two 16 bit registers. SELØ L selects the BAR while SEL2 L selects the WCR. INWD L reads the selected register,

and OUTHB L and OUTLB L write the high and low byte into the selected register.

Data is transferred through the read only data input register (DATIR-74LS374) and the write only data output register (DATOR-74LS273). CHANHB H and CHANLB H are used to write data into the high and low byte of the DATOR, and DATEN L is used to read data from the DATIR. Under program control DATEN L is generated by SEL6 L and INWD L, and during a DMA DATO or DATIO cycle by DATN H. Also under program control CHANHB H and CHANLB H are generated by SEL6 L and MRPLY L together with OUTHB L and OUTLB L; whereas, during a DMA DATI or DATIO cycle CHANHB H and CHANLB H are both generated by TDIN H.

The control and status register (CSR), as its name implies, is used to control and monitor the status of the device. The CSR consists of 3-state drivers (74LS244) and D flip-flops (74LS74, 74LS175, DC003) that are incorporated in the logic throughout this section. The read and write operations of the CSR are governed by CSRRD L, CSRWHB H, and CSRWLB H. SEL4 L selects the CSR and INWD L generates CSRRD L; CSRWHB H and CSRWLB H are generated by OUTHB L and OUTLB L together with MRPLY L. A description of the function of each of the CSR bits is given in Table 1.

4.5 Master Control - Figures 18 and 19

This section handles the control lines to and from the Biomation 6500 as well as implementing the various configurations

established by the CSR. All lines from the 6500 are terminated with 150 ohms and go into line receivers (74LS244) for increased noise immunity; lines to the 6500 are through line drivers (74LS244) for the same reason.

Before discussing the master control section a brief description of the Biomation 6500 control structure is in order, for more information consult the "Biomation 6500 Operating Manual".[4] Before the 6500 may be triggered the trigger circuitry must be armed, this is done by asserting RMA L. After being armed the next trigger pulse initiates the record cycle indicated by RCD L going low. After the analog waveform has been recorded RCD L is negated. OPT L must then be asserted within 10 microseconds from negation of RCD L to enter the digital output mode. In the digital output mode the two handshape control lines FLG H and WDC L are used to output the 1024 words of stored information. FLG H is asserted to indicate that valid data is present on the 6500's output lines. The next word is then fetched when WDC L is asserted which negates FLG H for 350 ns (6500's fetch time). This continues until all of the memory locations have been accessed at which time OPT L is negated and the 6500 is rearmed by RMA L and the cycle repeats.

The signal averaging (SA) cycle begins by setting the start bit of the CSR, which in turn asserts DEVMAS L if the attention bit of the CSR is not set. DEVMAS L enables the

line drivers and receivers (74LS244) for RCD L, OPT L, and RMA L. DEVMAS L also triggers a 1 microsecond one-shot which times out and triggers a 150 ns one-shot to produce RMA L. After being armed the next trigger pulse at the 6500 begins the record cycle signified by assertion of RCD L. RCD L also resets the arm flip-flop. When the record cycle is complete RCD L is negated which sets the SA cycle flip-flop to generate SACYC H and OPT L. The negation of RCD L also increments the SA scans counter (74LS393) which is preset to a count of one by the application of INIT H. SACYC L enables the line driver and receiver for FLG H and WDC L. The 6500 responds to OPT L by asserting FLG H to signify that the first word has been placed on the 6500 data bus. Approximately 100 ns after FLG H goes high DSTBHB H and DSTBLB H are generated to latch the data into the DATOR. Either DSTBHB H or DSTBLB H clocks the DZERO flip-flop which negates DZERO L and triggers a 200 ns one-shot to produce WP L which writes the resultant sum into the SA memory. At the same time WCD L is also asserted to fetch the next word from the 6500. At the end of WP L MACLK L is asserted which loads the next address into the MAR and also increments the MAC. The beginning of the 250 ns pulse also triggers a 400 ns one-shot which reasserts DZERO L and holds off acknowledgement of FLG H to allow for the 6500's data lines to settle and allow for the next SA memory location to be accessed. After the 400 ns one-shot times out, FLG H is honored and the

cycle repeats until RESET L is generated from the MAC. RESET L negates OPT L and SACYC H and clocks the arm flip-flop which asserts READY H to indicate that the device is ready to accept more data, and produces the 150 ns RMA L pulse after a 1 microsecond delay to the 6500 to rearm the trigger circuitry. This permits a data output rate from the 6500 of ∿ 700 ns per word which amounts to a signal averaging repetition rate of ∿ 1-1.5 kHz for the faster time bases.

Scans are continuously processed until the scans counter, which is incremented after each record cycle, reaches 512 scans. At this point RMA L is inhibited from being reasserted, and at the end of the SA cycle NWOVF H is gated to generate REQ H which initiates a DMA cycle.

The results of averaging 512 scans from the 6500 is now stored in the SA memory and is ready to be transferred to the main memory of the LSI-11. At this point the user can configure the transfer in several different ways through the use of the CSR. The transfer can be configured as either a DATO or DATIO though the use of the DATIN and DATIO CSR bits. For a DATO transfer the contents of the SA memory can be reset to zero or left unchanged after the transfer by setting the destructive bit of the CSR. For a DATIO transfer the resultant word size in the main memory can be either 16 or 32 bits corresponding to either single or double precision integer

format. The contents of the SA memory are automatically reset to zero during a DATIO transfer.

In the normal SA operating mode a DATIO transfer would be performed with a 32 bit expansion of the stored 16 bit SA memory contents. In this configuration the DATIO bit of the CSR is set which asserts DATIO H, and the one word transfer bit is not set so that OWT H is set low. When the first address is placed on the bus ADREN L is asserted which toggles the double word flip-flop which enables the sign bit (SB) and carry (COUT) latches to be loaded by SUMLD L. The 200 ns SUMLD L pulse is generated by CHANHB H or CHANLB H in conjunction with DIN H which also produces DSTBHB H and DSTBLB H to load the DATOR. SUMLD L is then gated to produce DATLD H which loads the DATIR with the sum of the SA memory and the main memory. The DATIR is then written back into the same location in main memory by DATEN L. The next address is then placed on the bus by ADREN L which toggles the double word flip-flop back again. This inhibits the SB and COUT latches and gates their outputs to produce SUMCLR L and CIN and also negates CS L. If a carry was produced by the previous addition then CIN is set high, otherwise CIN is low. Likewise, if the SB was high for the previous addition, indicating a negative number, then SUMCLR L is not asserted; otherwise SUMCLR L is asserted.

DATOR is then loaded by DSTBHB H and DSTBLB H with the contents of the second main memory location. This is then

added to the contents of the SA MD bus which is either all zeros (positive add) or all ones (negative add) together with the carry from the previous addition. This result is then loaded into the DATIR by DATLD H and is written back into the same location in main memory by DATEN L to form the second half (upper 16 MSBs) of the 32 bit expanded word. On the second word DATEN L is gated through to produce SUMCLR L, SUMEN L and CS L. At the same time, it triggers the 200 ns one-shot which in turn triggers the 250 ns one-shot to produce WP L and MACLK L which writes zero into the current SA memory location and increments the MAC to the next location. This continues until all of the SA memory locations have been transferred and the DATIO cycle terminates.

For single precision transfers OWT H is asserted so that the double word flip-flop is inhibited from being toggled by ADREN L. Thus the sign bit and carry are ignored and the sum of the SA memory and main memory is truncated to 16 bits so that only 1024 words of main memory are needed for storage. Note that overflows and underflows that may occur with single precision transfers are not corrected for during the transfer.

If a DATO transfer is requested, setting the destructive bit of the CSR causes DEST L to be asserted which enables WP L to be asserted. DATEN L is used to generate DATLD H which

loads the DATIR. Since DZERO L is set the A adder input is zero, and thus the DATIR is loaded with the unaltered contents of the SA memory. DATEN L also asserts SUMCLR L and SUMEN L, and triggers a 200 ns one-shot which times out and triggers the 250 ns one-shot to produce WP L and MACLK L. Since SUMCLR L and SUMEN L were both asserted this writes zero into the SA memory location from which the DATIR was loaded. If the destructive bit of the CSR is not set then DEST L is negated and inhibits WP L. Thus the contents of the SA memory are left unchanged during the transfer.

The SA may also be configured for a DATI cycle to transfer data from the LSI-11 to the SA. This can be used, for example, to display on a scope acquired data that has been subsequently manipulated on by a user program; or simply as a fast temporary storage area utilized by a user routine. Through the use of the CSR destructive bit it is possible to configure the SA memory so that incoming information either overwrites or adds to the present SA memory contents.

CHANHB H and CHANLB H are used in conjunction with TDIN H to generate DSTBHB H and DSTBLB H to latch the incoming data from the LSI-11 bus into the DATOR. DSTBHB H or DSTBLB H triggers the 200 ns SUMLD L pulse. If DEST L is asserted, SUMCLR L is also asserted so that the contents of the DATOR are added to zero. If DEST L is negated, SUMCLR L is not asserted and the contents of the DATOR are added to the contents

of the SA memory. The trailing edge of SUMLD L clocks the 250 ns one-shot to produce WP L and MACLK L which writes the result into the SA memory and clocks the MAC to the next memory location. This continues until all the words are transferred and the DATI cycle terminates.

The SA memory may also be erased by setting the erase bit of the CSR which asserts ERASE H. When ERASE H goes high it assets SUMCLR L, SUMEN L and CS L which sets the SA MD bus to zero. ERASE H also triggers SUMLD L which triggers the 250 ns one-shot to produce WP L and MACLK L. This writes zero into the current SA memory location and increments the MAC to the next location. A 400 ns one-shot is also triggered at the beginning of WP L which allows ERASE H to retrigger SUMLD L and repeat the above sequence of writing zero to memory. This continues until all of the memory locations are cleared at which time RESET L is generated by the MAC and negates ERASE H to terminate the erase cycle. The total erase cycle time is < 1 millisecond.

5. CONCLUSIONS

Currently the SA interface is being used with an LSI-11/2 CPU using the RT-11 V3B operating system. The total system configuration is depicted in Figure 20. Various MACRO and FORTRAN routines have been written for the device which provide most of the commonly needed features, such as: SA handler, SA command interpreter, addition and subtraction of waveforms, multiple waveform storage, integration, differentiation, FFT, curve fitting, extended graphics package, diagnostics, and others as dictated by the user's needs. The SA command interpreter allows all of the above features to be accessed interactively through the keyboard via simple four letter mnemonics. Additional commands can be readily incorporated into the SA command interpreter as the user's needs change. Thus one has at their disposal a system that is both powerful and easy to use.

Table I. CSR OPERATIONAL STATUS

BIT	DESCRIPTION (H=1, L=Ø)
15	Sets H when (1) bus does not reply 10 μs after addressed (non-existent address - prevents bus hangup), (2) attention is called for (not device originated) and a SA cycle or DMA cycle are not being processed (if so it sets at cycle completion). Normally attention is called by operator intervention to halt the device without destroying current contents. For both (1) and (2) bit 14 must be set H to enable "A" interrupt.
14	Must be set H to enable bit 15 - "A" interrupt request. Otherwise "A" interrupt is ignored.
13	Sets H when word count overflows signifying the end of the DMA cycle - generates interrupt request "B".
12	Must be set H to enable interrupt request "B", otherwise "B" interrupt is ignored.
11	Sets H when the device requires a DMA;resets back L on word count overflow (also on non-existent address and INIT).
10	Determines R/W status of SA memory. H indicates DOUT cycle (device to bus), L indicates DIN cycle (bus to device).
9	Determines R/M/W status of SA memory if set L causes DATIO cycle.
8	Resets device when set H (1 μs RESET-INIT pulse).
7	Erases SA memory when set H (erase time ∿ 1 ms). Resets to L at end of ERASE cycle.

BIT	DESCRIPTION (H=1, L=Ø)
6	If set H causes contents of memory to be clocked at ∿ 1 μs/word - used to display memory contents on X-Y scope - contents can be either SA result or data returned from the bus.
5	Extend memory address bit 17 used to expand addressable memory from 32k words to 128k words - used by MMU on LSI-11/23.
4	Extended memory address bit 16 - see bit 5.
3	When set H inhibits 16 bit → 32 bit increase in word size during DOUT cycle - single word precision. When set L causes 32 bit storage of data - double word precision (I*4 format).
2	When set L causes destructive read of memory (during DOUT clears after read, during DIN overwrites) when set H causes non-destructive readout. (During DOUT simple read, during DIN does an add to SA memory). No effect on DATIO cycle.
1	When set H starts the device.
Ø	When H causes "A" interrupt to be set at end of current SA or DMA cycle - stops device.

References

1. L. T. Specht, "A Computer-Based High Speed Digital Signal Averager, M.S. Thesis, University of Illinois, May 1977.

2. LSI-11 Microcomputer Handbook, Digital Equipment Corporation, Maynard, MA, 1978.

3. CHIPKIT User's Manual, Digital Equipment Corporation, Maynard, MA, 1979

4. Biomation 6500 Operating Manual, Biomation, Santa Clara, CA, 1978.

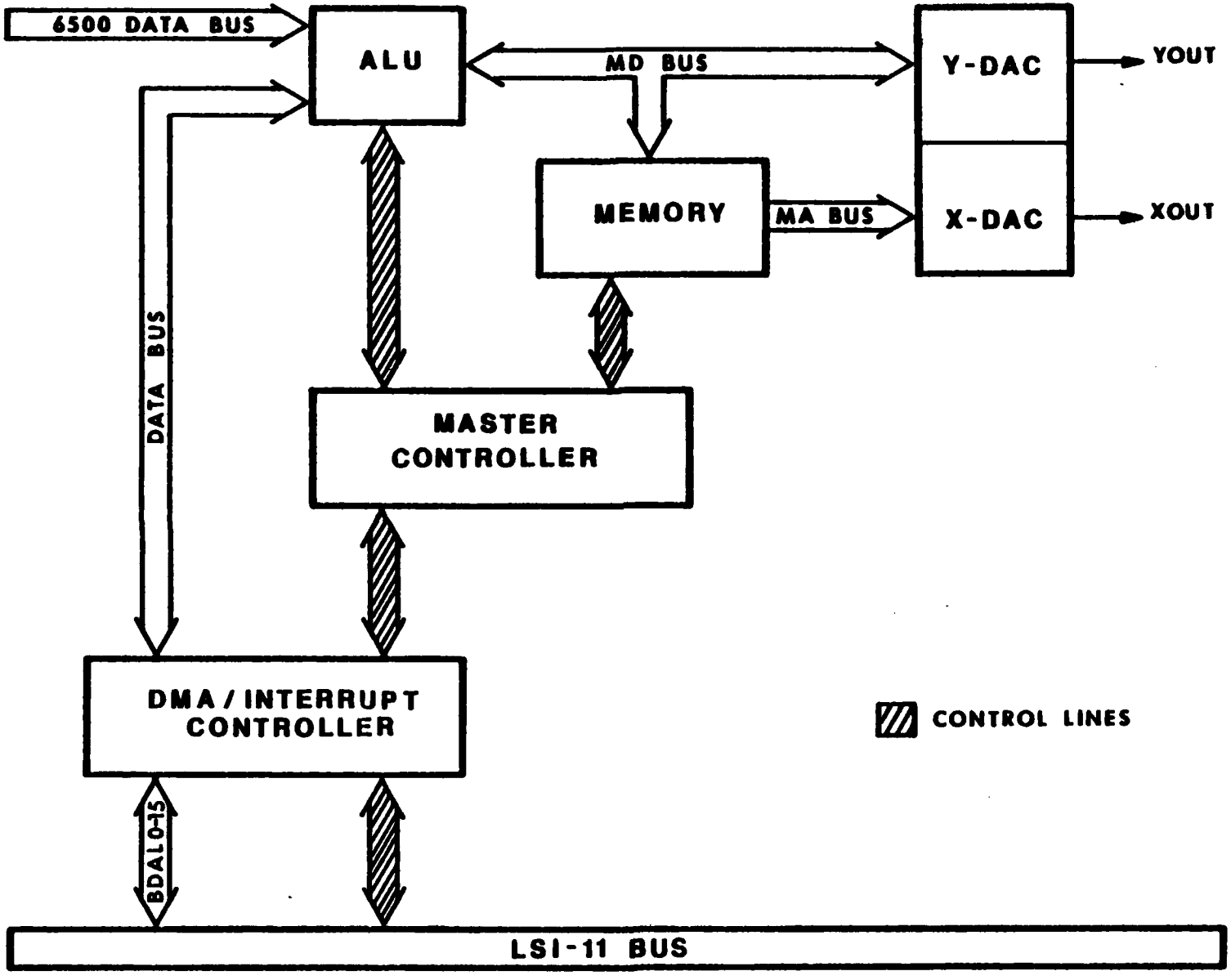


Figure 1. Signal averager interface block diagram.

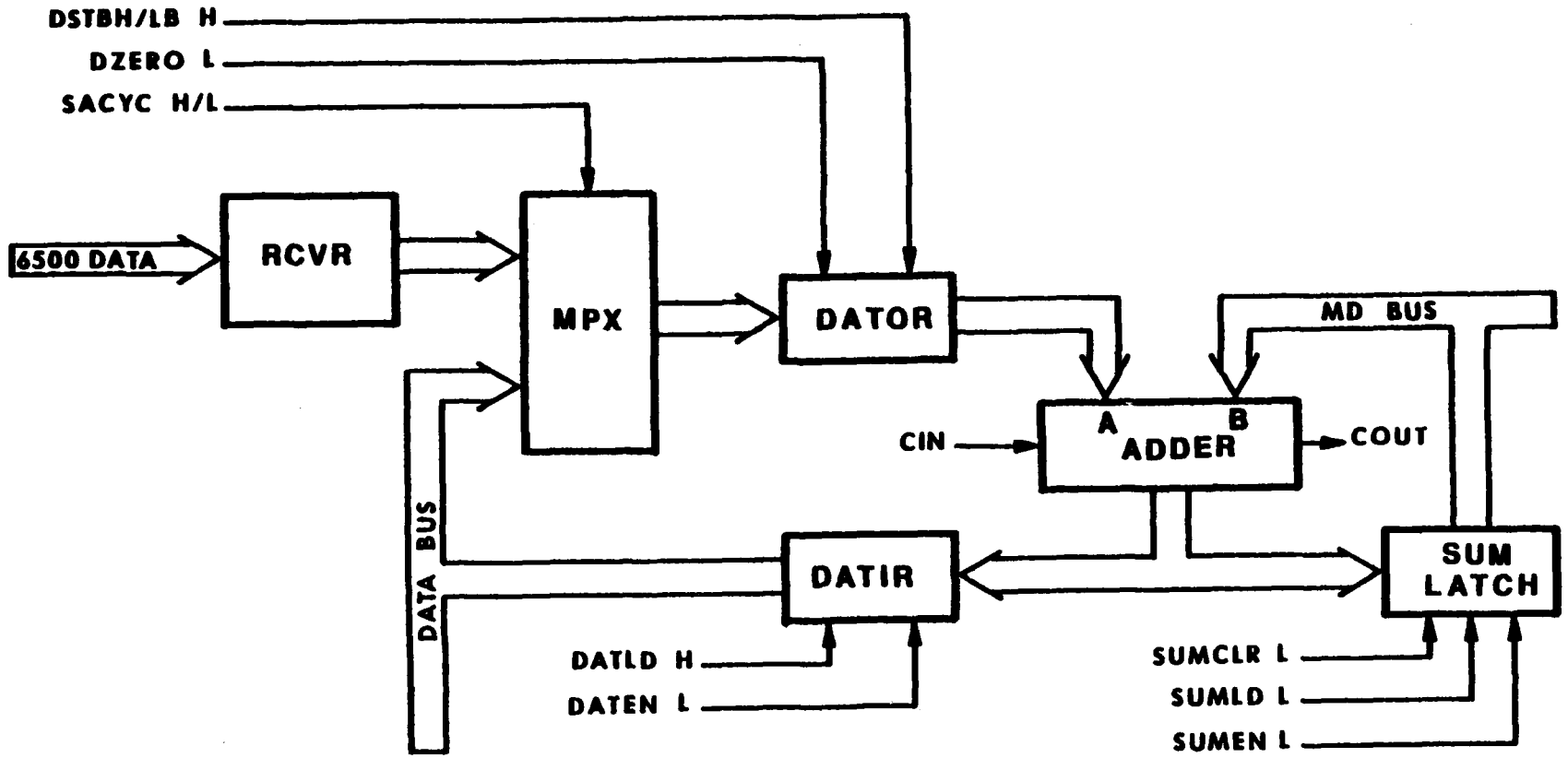


Figure 2. Arithmetic logic unit section.

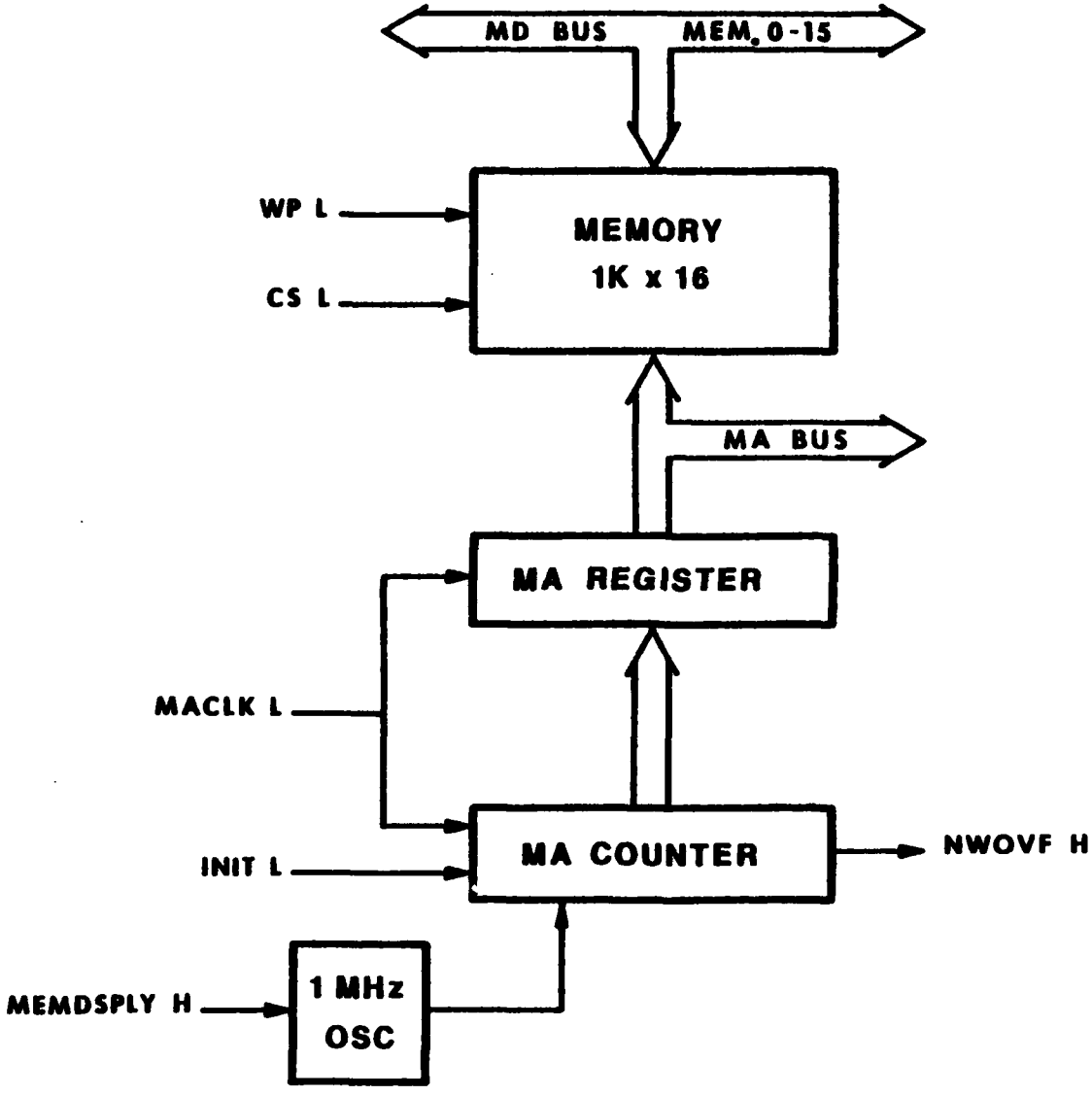


Figure 3. Memory section.

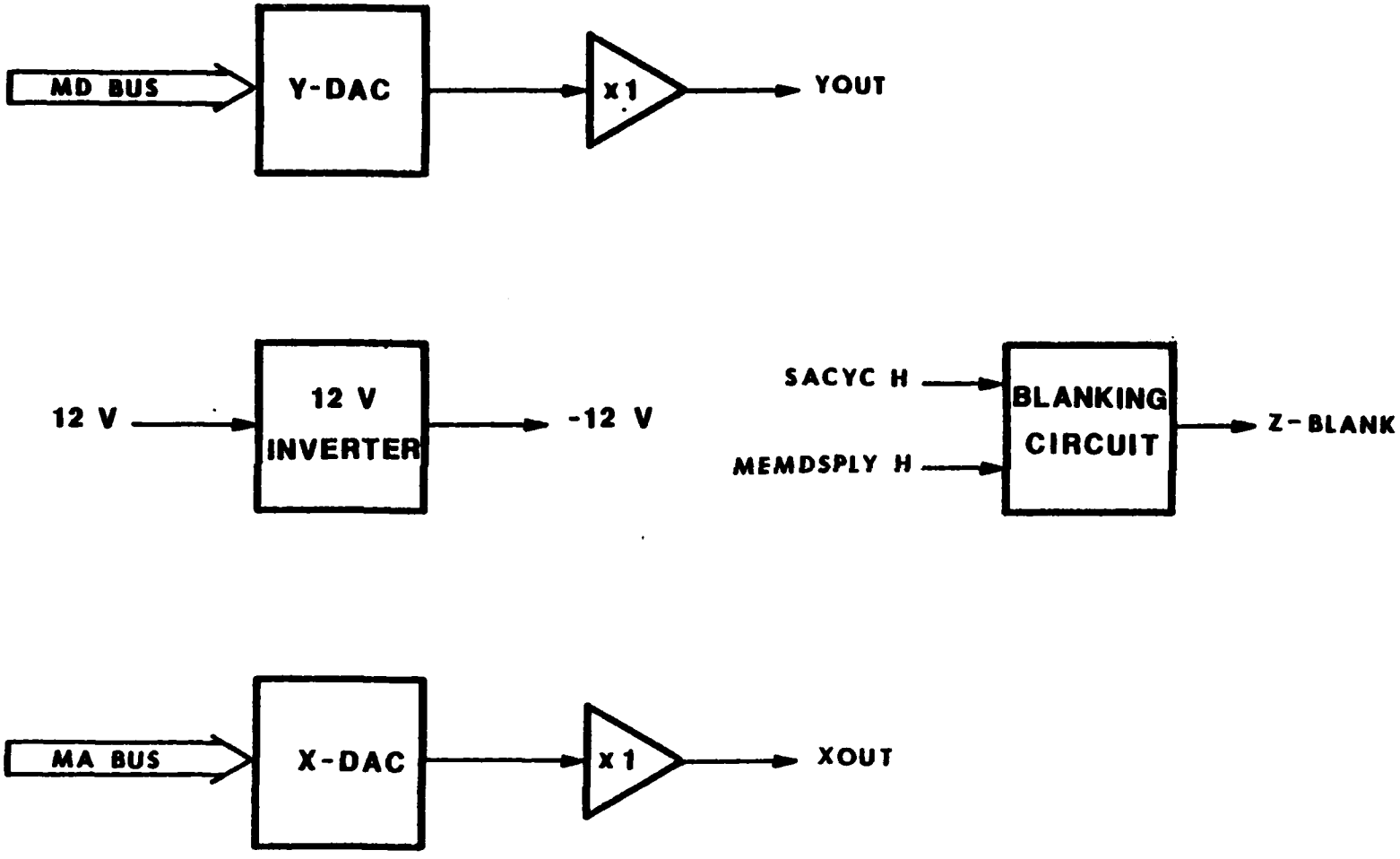


Figure 4. x and y digital-to-analog converter section.

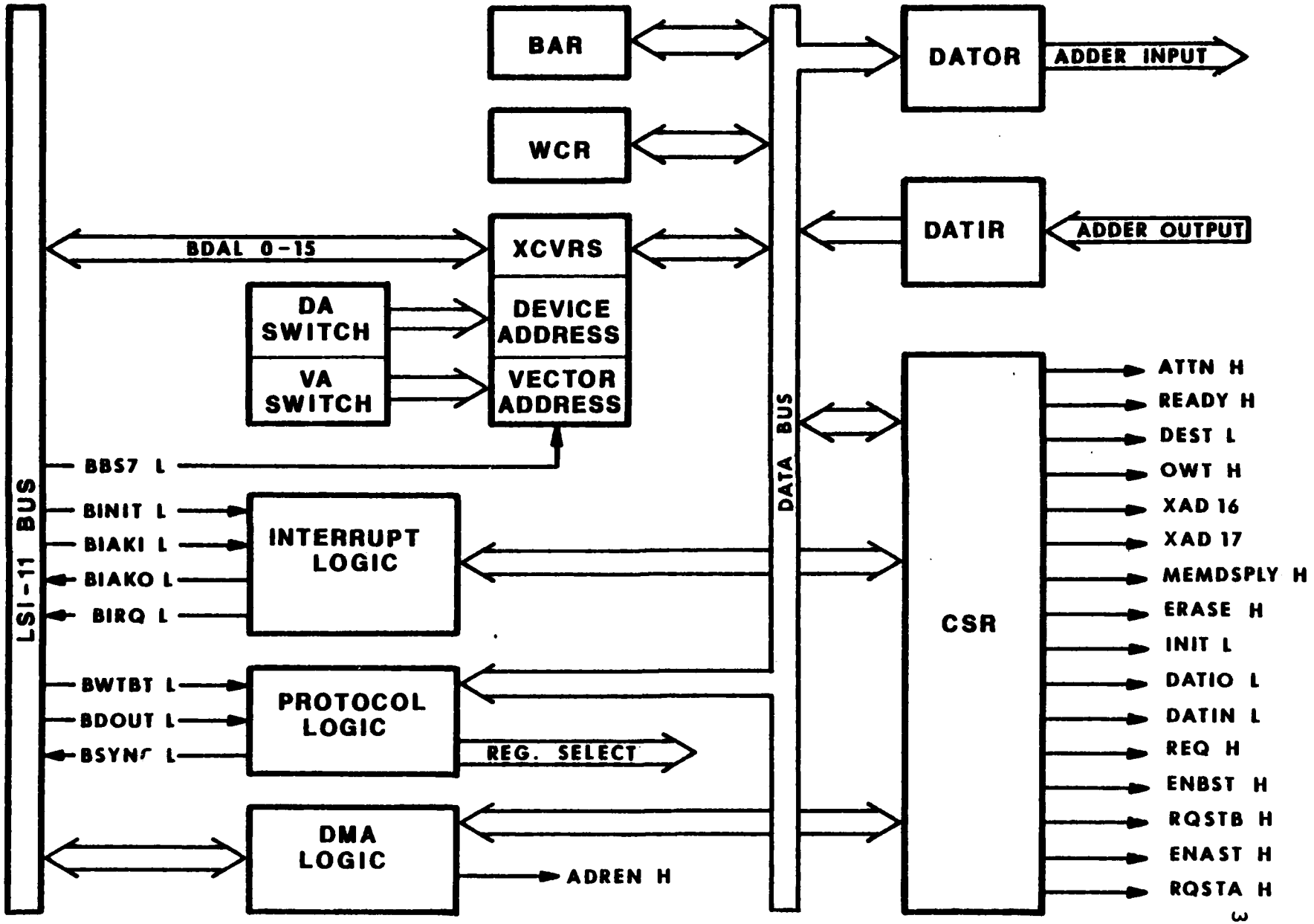


Figure 5. Direct memory address and interrupt section.

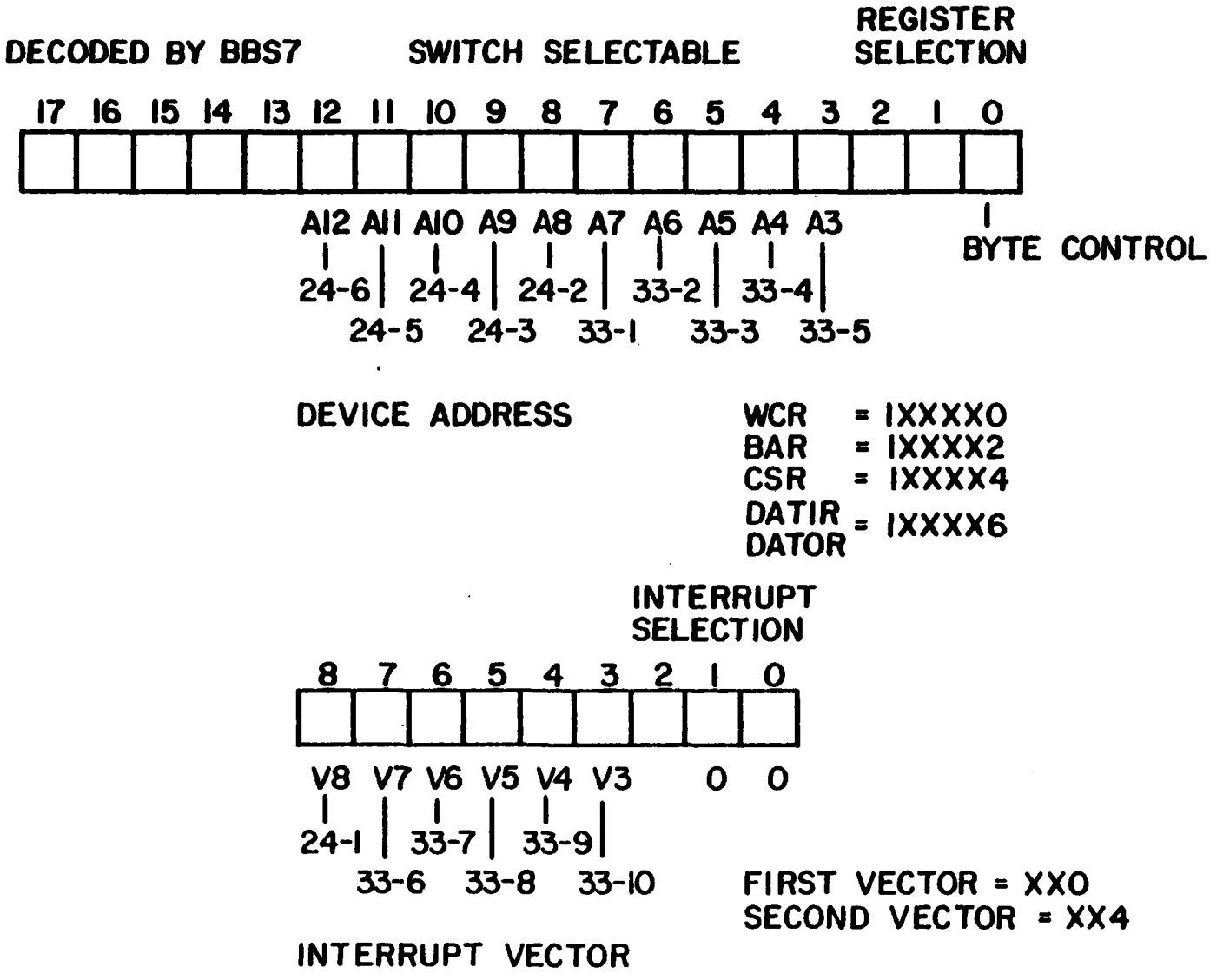


Figure 6. Device address and vector address format.

```
;       PRELIMINARY SIGNAL AVERAGER UTILITY ROUTINES
;
;       LTS - 11-JUN-79
;
        .MCALL  ..V2..,.REGDEF
        .MCALL  .PRINT,.DEVICE,.PROTECT,.INTEN,.MFPS,.MTPS
        ..V2..
        .REGDEF
        .GLOBL SAINIT,SABUF,SADPLY,SAERSE,SARSET,SABEGN
        .GLOBL SASTAT,SAFILL,SASTOP,SASTRT,SASCAN,SAHALT
;
SA$CSR=176534
SA$VEC=330
PRI7=340
PRI5=5
;
SABEGN: .DEVICE #AREA,#DLIST
        .PROTECT#AREA,#SA$VEC
        BCS     2$
        .PROTECT#AREA,#SA$VEC+4
        BCS     2$
        MOV     #SAIH,@#SA$VEC
        MOV     #PRI7,@#SA$VEC+2
        MOV     #SAIH,@#SA$VEC+4
        MOV     #PRI7,@#SA$VEC+6
        CCC
2$:     RTS     PC
AREA:   .BLKW   10.
DLIST:  .WORD   SA$CSR,400,0
SADPLY: MOV     @#SA$CSR,SACSR
        JSR     PC,SARSET
        BIS     #000100,@#SA$CSR
        RTS     PC
SAERSE: MOV     @#SA$CSR,SACSR
        JSR     PC,SARSET
        BIS     #000200,@#SA$CSR
1$:     BIT     #000200,@#SA$CSR
        BNE     1$
        JSR     PC,SARSET
        MOV     SACSR,@#SA$CSR
        RTS     PC
SARSET: BIC     #050000,@#SA$CSR
        BIS     #000400,@#SA$CSR
1$:     BIT     #000400,@#SA$CSR
        BNE     1$
        RTS     PC
SAFILL: MOV     @#SA$CSR,SACSR
        JSR     PC,SARSET
```


Figure 7. A simple interrupt handler routine.

```
        MOV     R0,-(SP)
        TST     (R5)+
        MOV     (R5)+,R0
        DEC     R0
1$:     MOV     @(R5)+,@#SA$CSR
        DEC     R0
        BNE     1$
        MOV     (SP)+,R0
        JSR     PC,SADPLY
        RTS     PC
SASTAT: MOV     R2,-(SP)
        MOV     @#SA$CSR,SACSR
        MOV     #MSGBLK,R2
        CLR     STATB
        SEC
3$:     ROL     STATB
        BCS     5$
        .PRINT  R2
        BIT     STATB,SACSR
        BNE     1$
        .PRINT  #MSG1
        BR      2$
1$:     .PRINT  #MSG2
2$:     ADD     #22,R2
        BR      3$
5$:     MOV     (SP)+,R2
        RTS     PC
STATB:  .WORD   0
MSG1:   .ASCIZ  /    NOT SET/
MSG2:   .ASCIZ  /    SET/
MSGBLK: .ASCII  /ATTENTION FLAG  /
        .WORD   100200
        .ASCII  /SA DEVICE START /
        .WORD   100200
        .ASCII  /DESTRUCTIVE READ/
        .WORD   100200
        .ASCII  /DOUBLE PRECISION/
        .WORD   100200
        .ASCII  /XM ADDRESS 16   /
        .WORD   100200
        .ASCII  /XM ADDRESS 17   /
        .WORD   100200
        .ASCII  /MEMORY DISPLAY  /
        .WORD   100200
        .ASCII  /MEMORY ERASE    /
        .WORD   100200
        .ASCII  /SA DEVICE RESET /
        .WORD   100200
        .ASCII  /READ-MOD-WRITE  /
        .WORD   100200
        .ASCII  /DATA READ-OUT   /
```

Figure 7. Continued.

```
        .WORD   100200
        .ASCII  /DMA REQUEST     /
        .WORD   100200
        .ASCII  /DMA INTERRUPT EN/
        .WORD   100200
        .ASCII  /DMA INTERRUPT   /
        .WORD   100200
        .ASCII  /ERR INTERRUPT EN/
        .WORD   100200
        .ASCII  /ERR INTERRUPT   /
        .WORD   100200
        .EVEN
SASTOP: MOV     @#SA$CSR,SACSR
        BIS     #000001,@#SA$CSR
        MOV     R4,-(SP)
        MOV     #300,R4
1$:     DEC     R4
        BNE     1$
        MOV     (SP)+,R4
        JSR     PC,SARSET
        RTS     PC
SASTRT: BIT     #120000,@#SA$CSR
        BNE     1$
2$:     BIS     #050000,@#SA$CSR
        BIS     #000002,@#SA$CSR
        RTS     PC
1$:     BIC     #120000,@#SA$CSR
        .PRINT  #MSGINT
        BR      2$
MSGINT: .ASCIZ  /INTERRUPT IS SET, POSSIBLE ERROR/
        .EVEN
SAINIT: TST     (R5)+
        MOV     @(R5)+,SASCAN
SAREG:  MOV     WC,@#SA$CSR-2
        MOV     SABUFP,@#SA$CSR-4
        MOV     CS,@#SA$CSR
        RTS     PC
SASCAN: .WORD   0
WC:     .WORD   173777
CS:     .WORD   002000
SACSR:  .WORD   0
SABUFP: .WORD   SABUF
SABUF:  .BLKW   4030
SAIH:
        .INTEN  PRIS
        TST     @#SA$CSR
        BMI     ATTN
        BIT     #20000,@#SA$CSR
        BEQ     RET
        JSR     PC,SARSET
        JSR     PC,SAREG
```

Figure 7. Continued.

```
        DEC     $ASCAN
        SEQ     $AHALT
        JSR     PC,$ASTRT
        RTS     PC
ATTN:   JSR     PC,$ARSET
        RTS     PC
MSGXXX: .ASCIZ  <15><12>/SCANS COMPLETE/<15><12>/X/
        .EVEN
RET:    BIS     #50000,@#SA$CSR
        RTS     PC
$AHALT: .MFPS
        .MTPS   #PRI7
        JSR     PC,$ARSET
        .PRINT  #MSGXXX
        .MTPS
        RTS     PC
        .END
```


Figure 7. Continued.

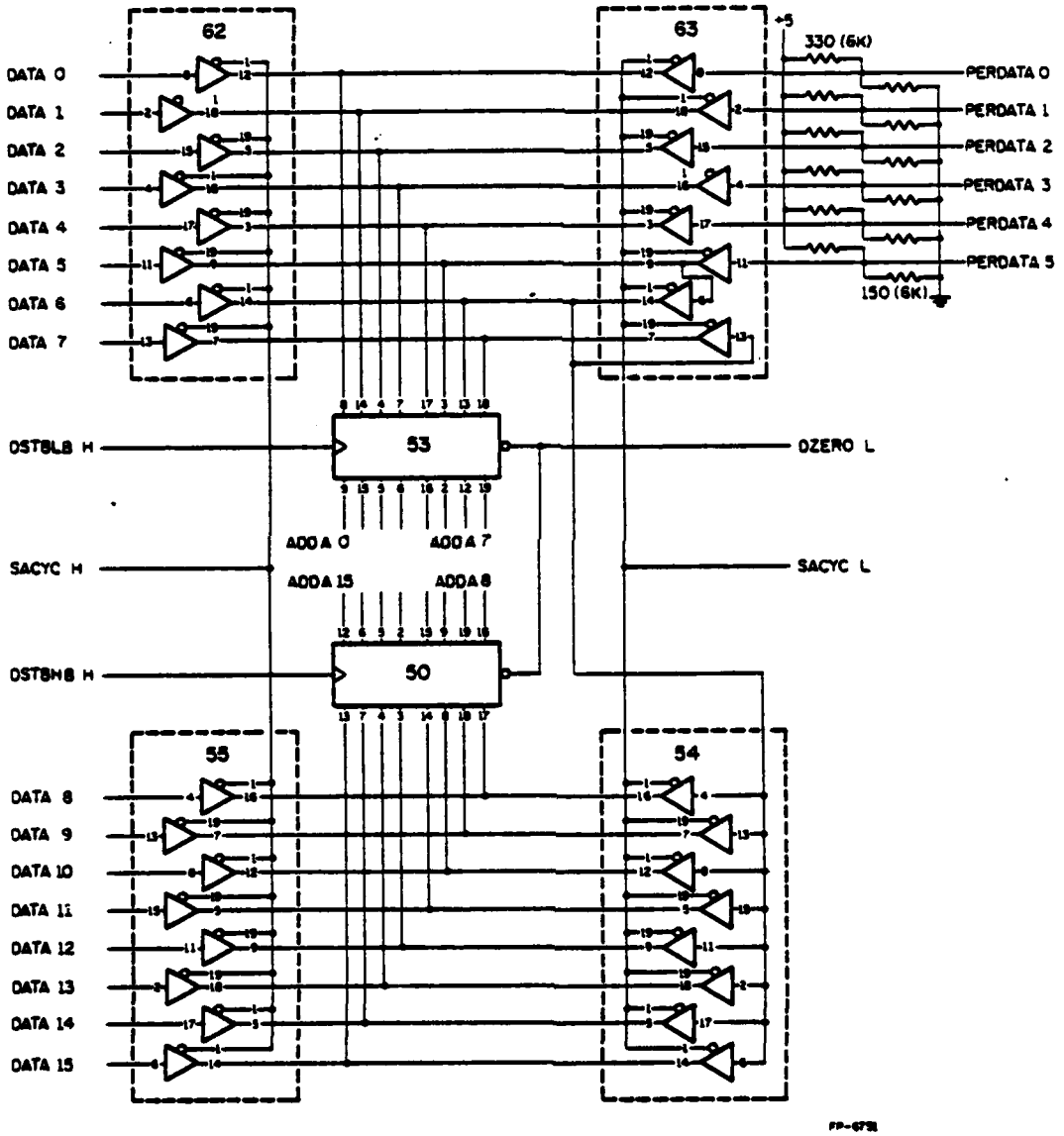


Figure 8. ALU schematics.

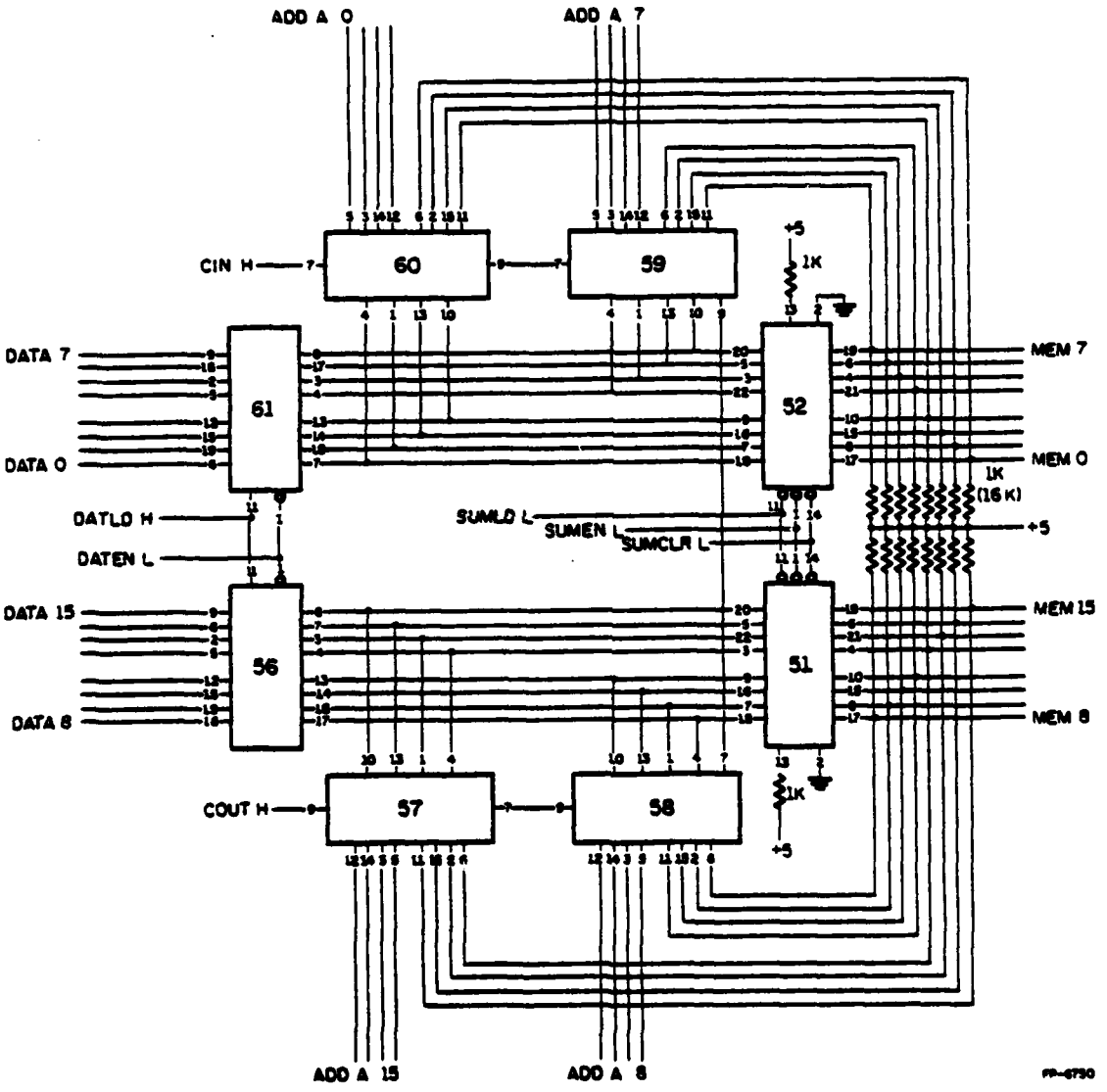


Figure 9. ALU schematics.

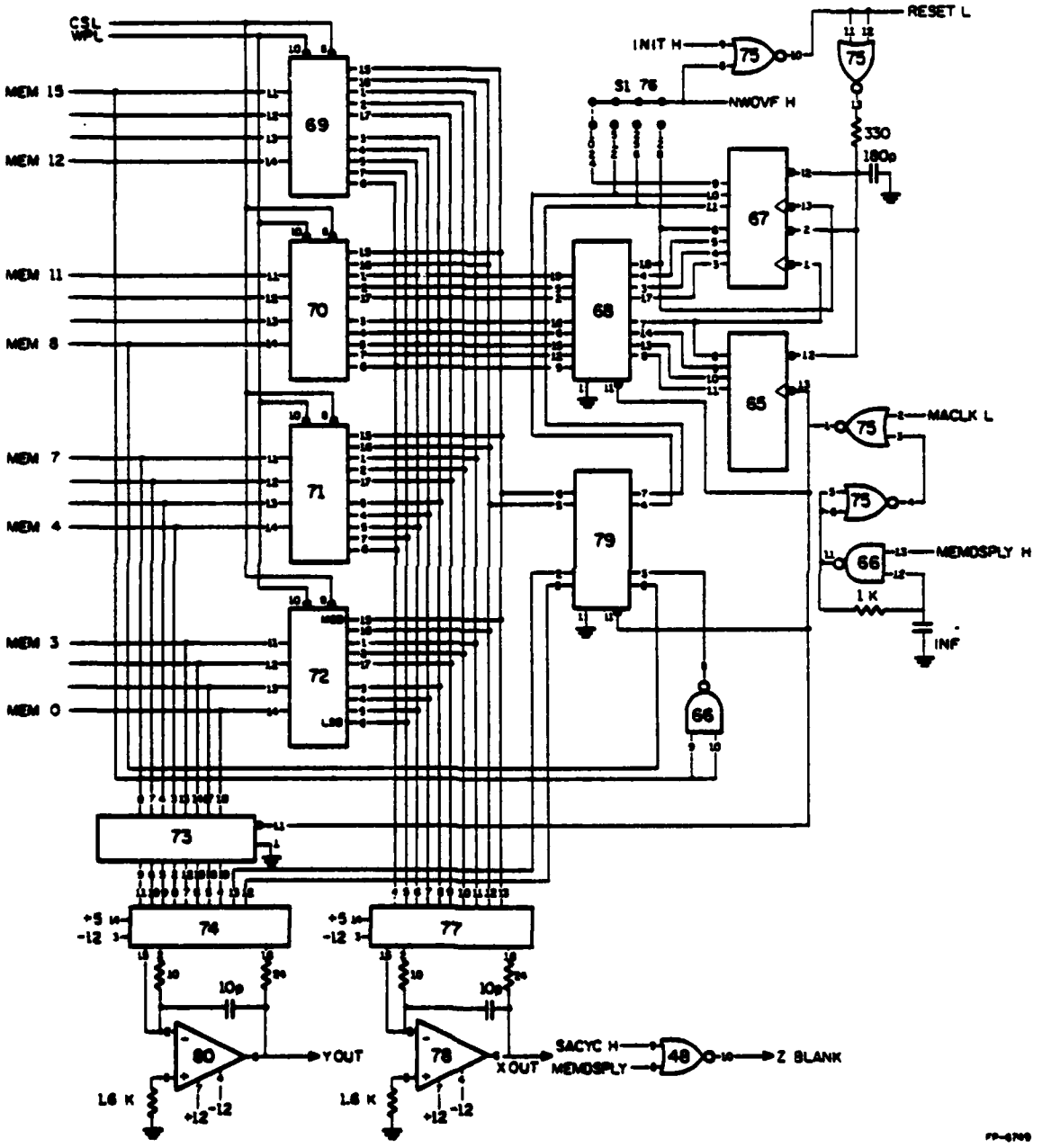


Figure 10. Memory and DAC schematic.

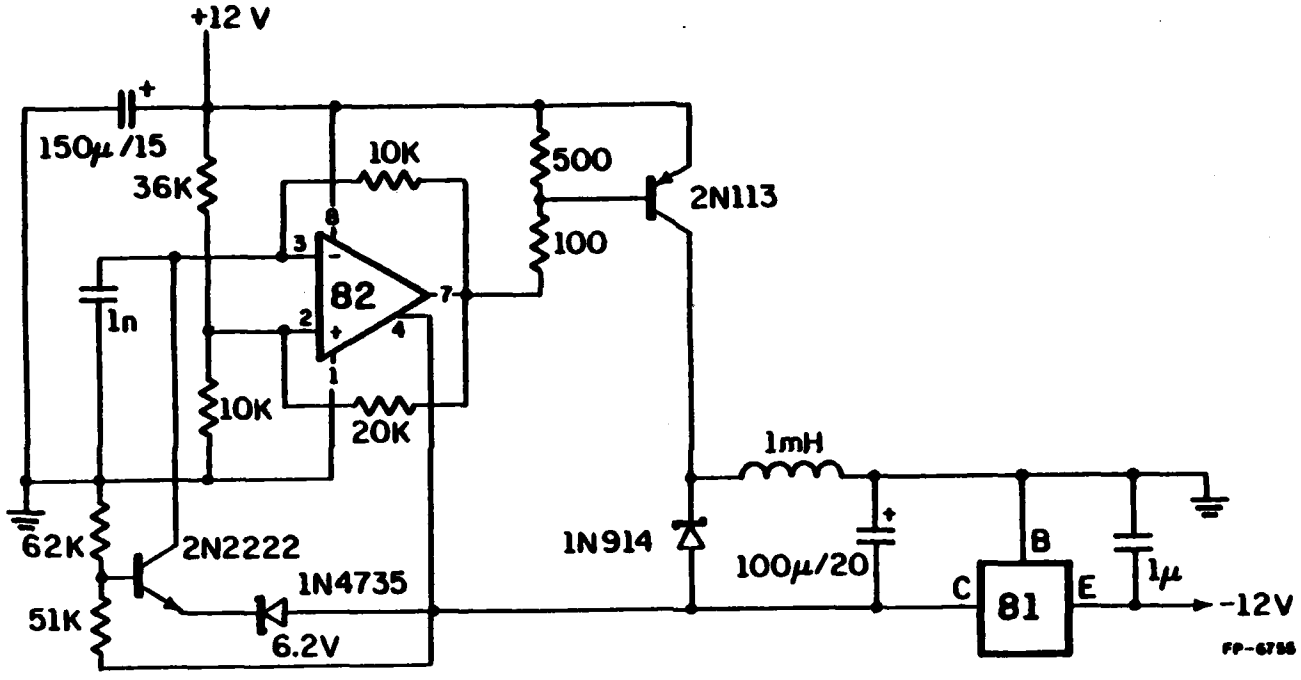


Figure 11. 12V inverter schematic.

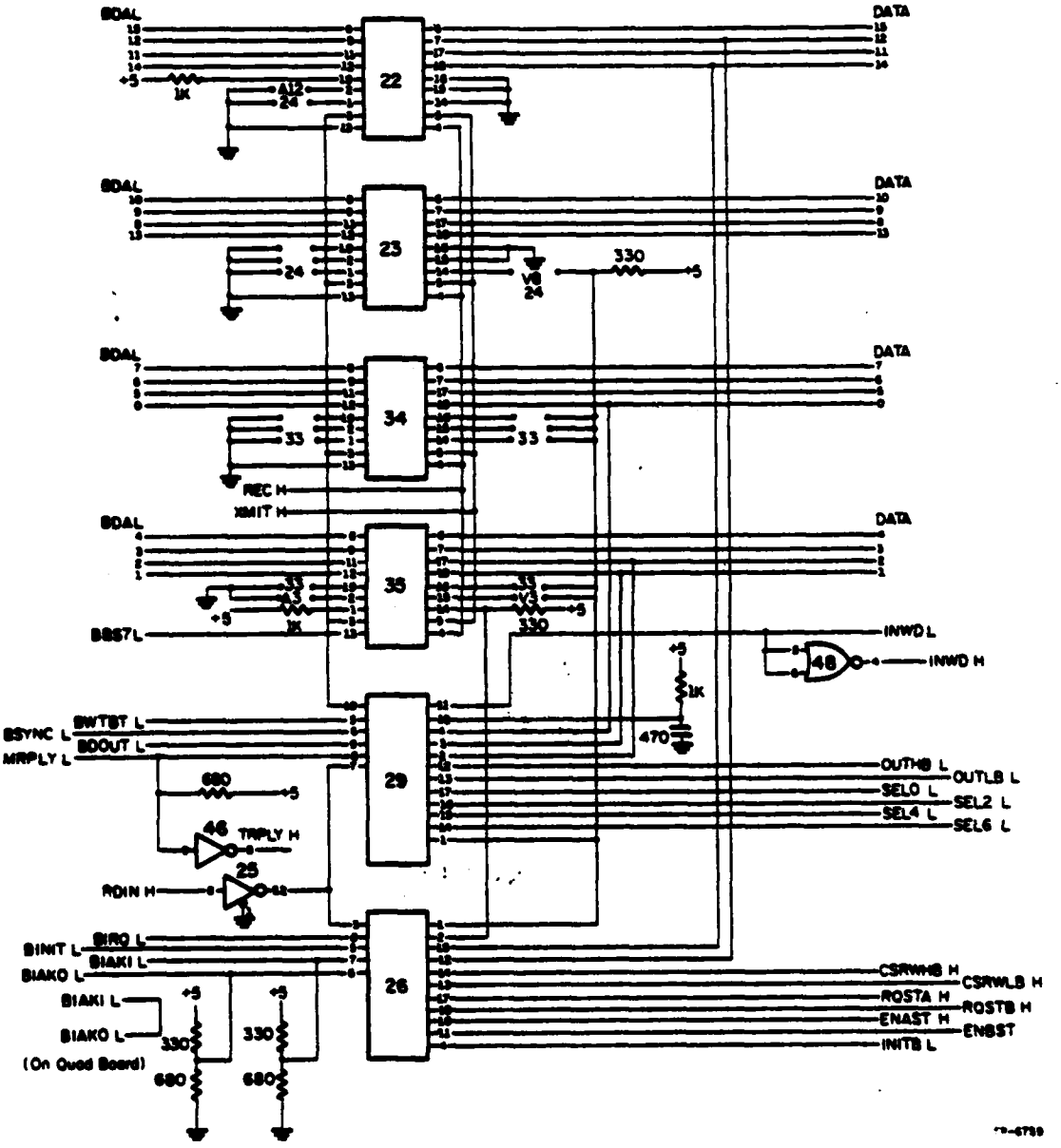


Figure 12. DMA and program control schematics.

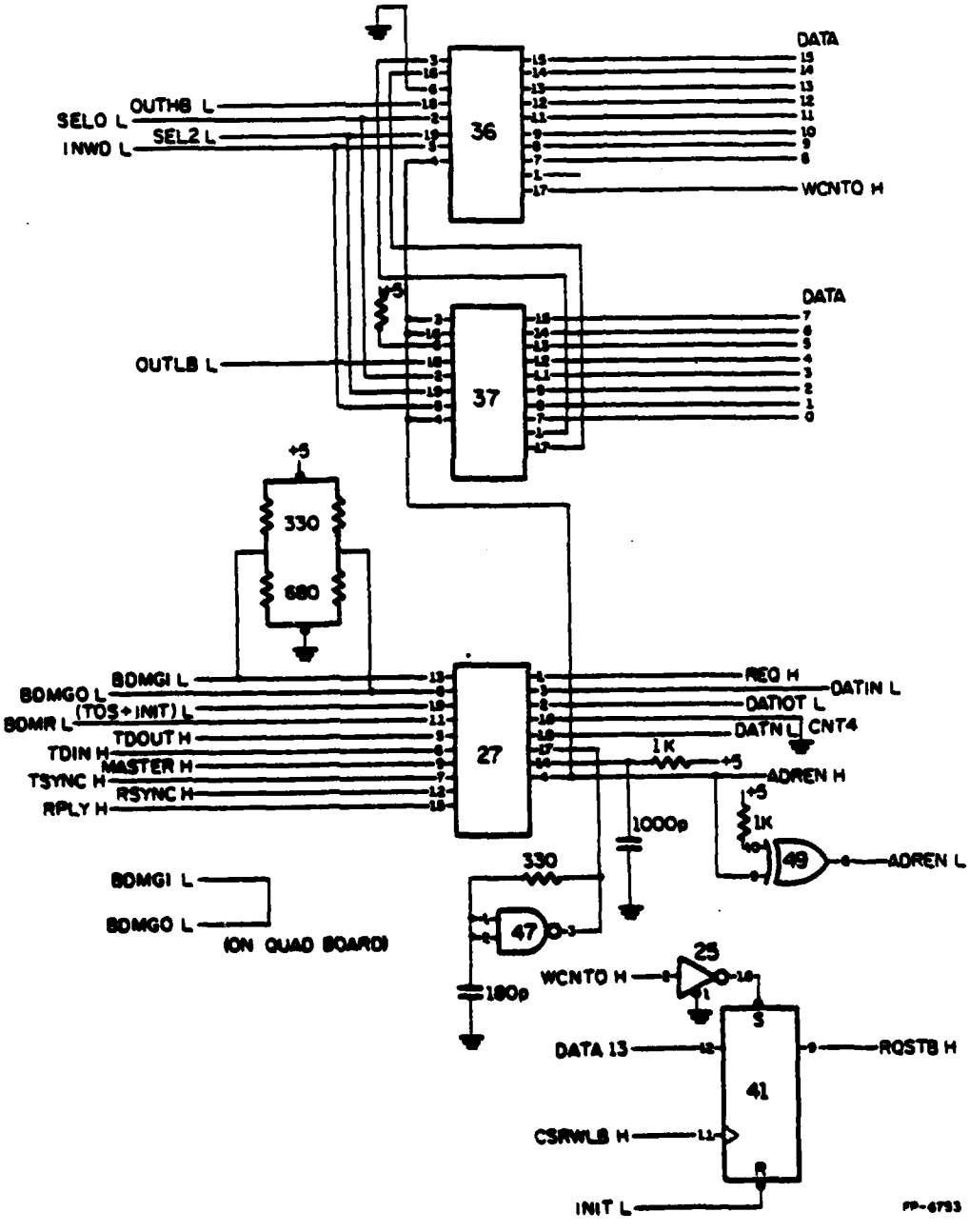


Figure 13. DMA and program control schematics.

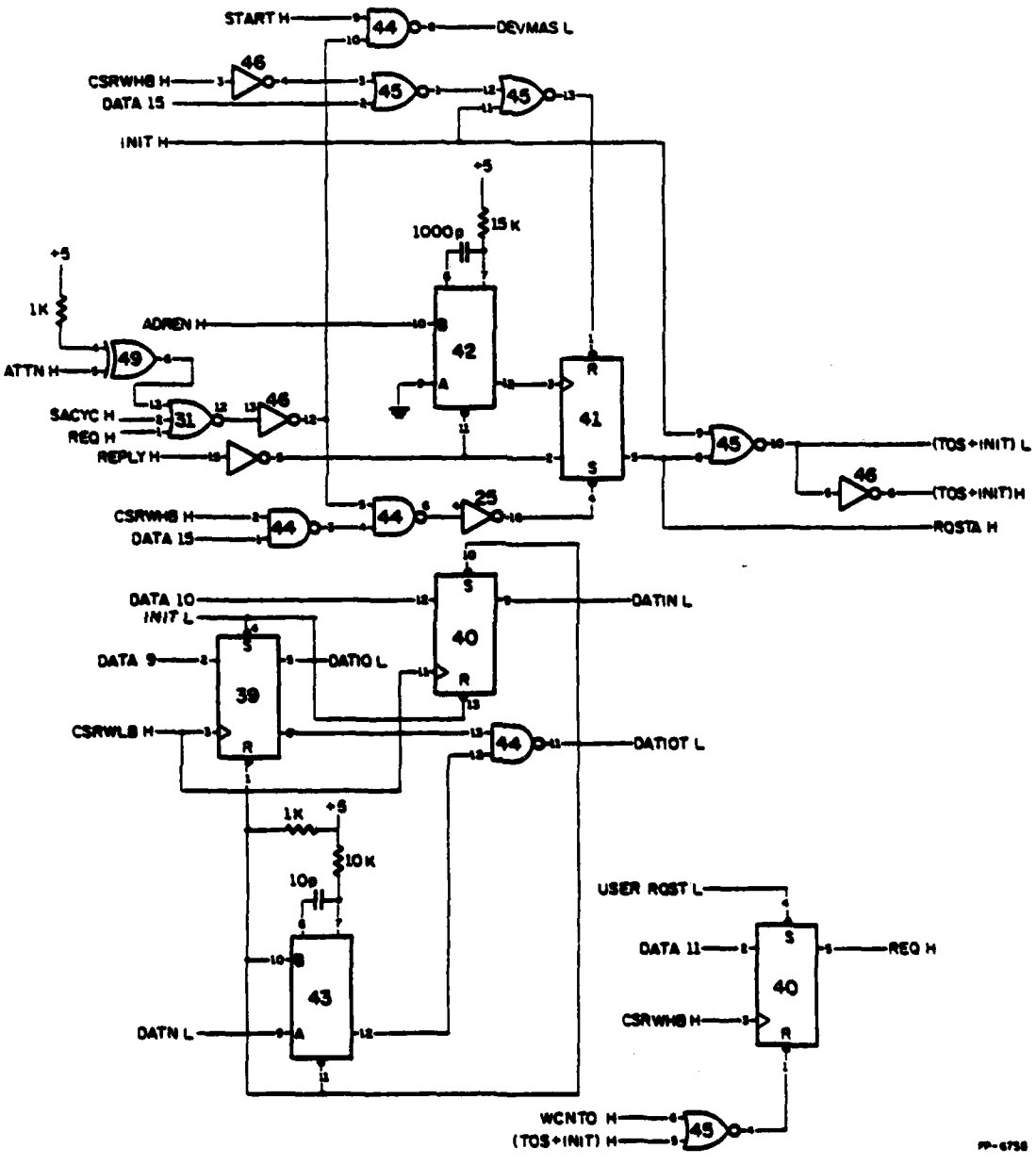


Figure 14. DMA and program control schematics.

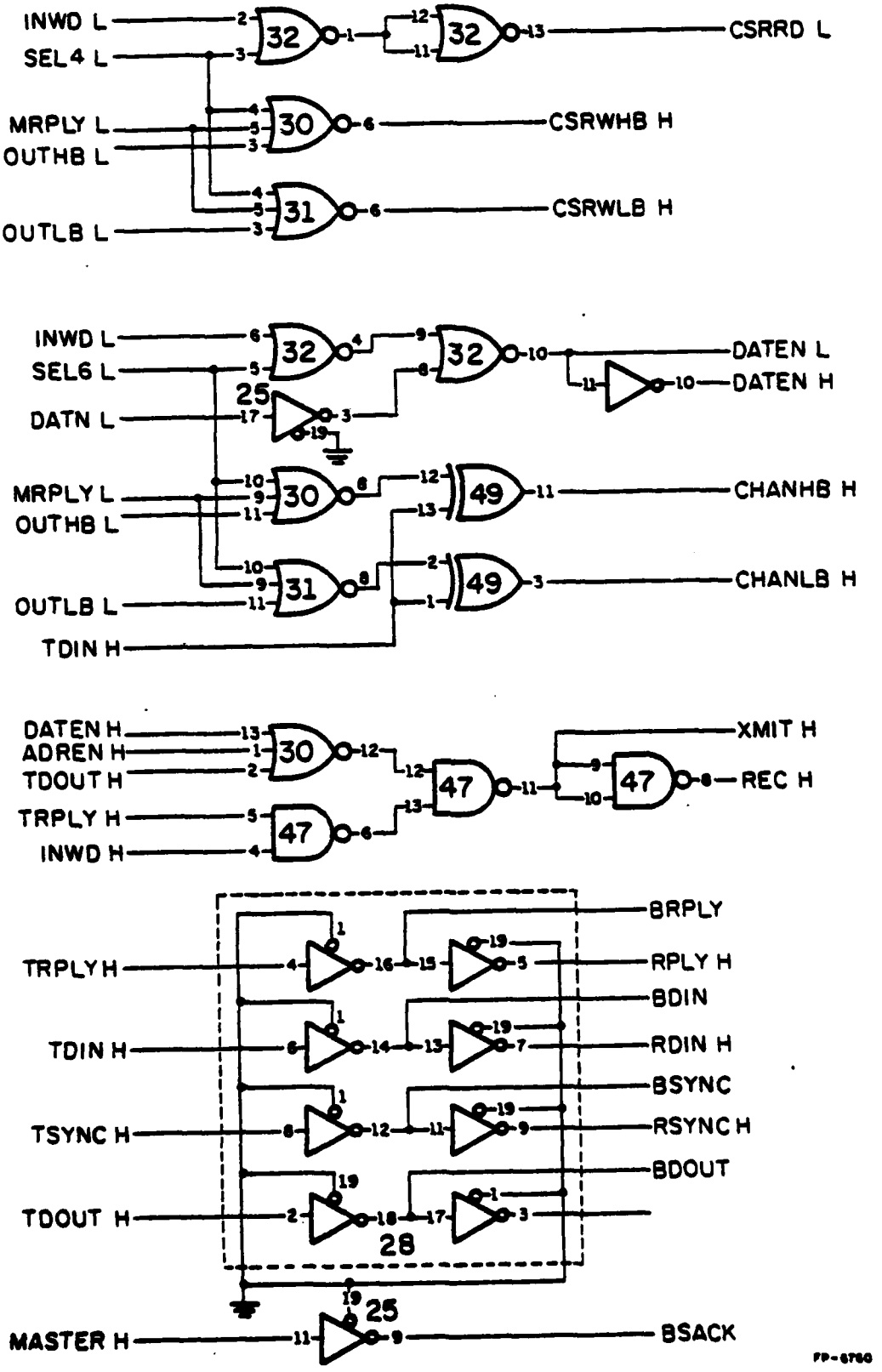


Figure 15. DMA and program control schematics.

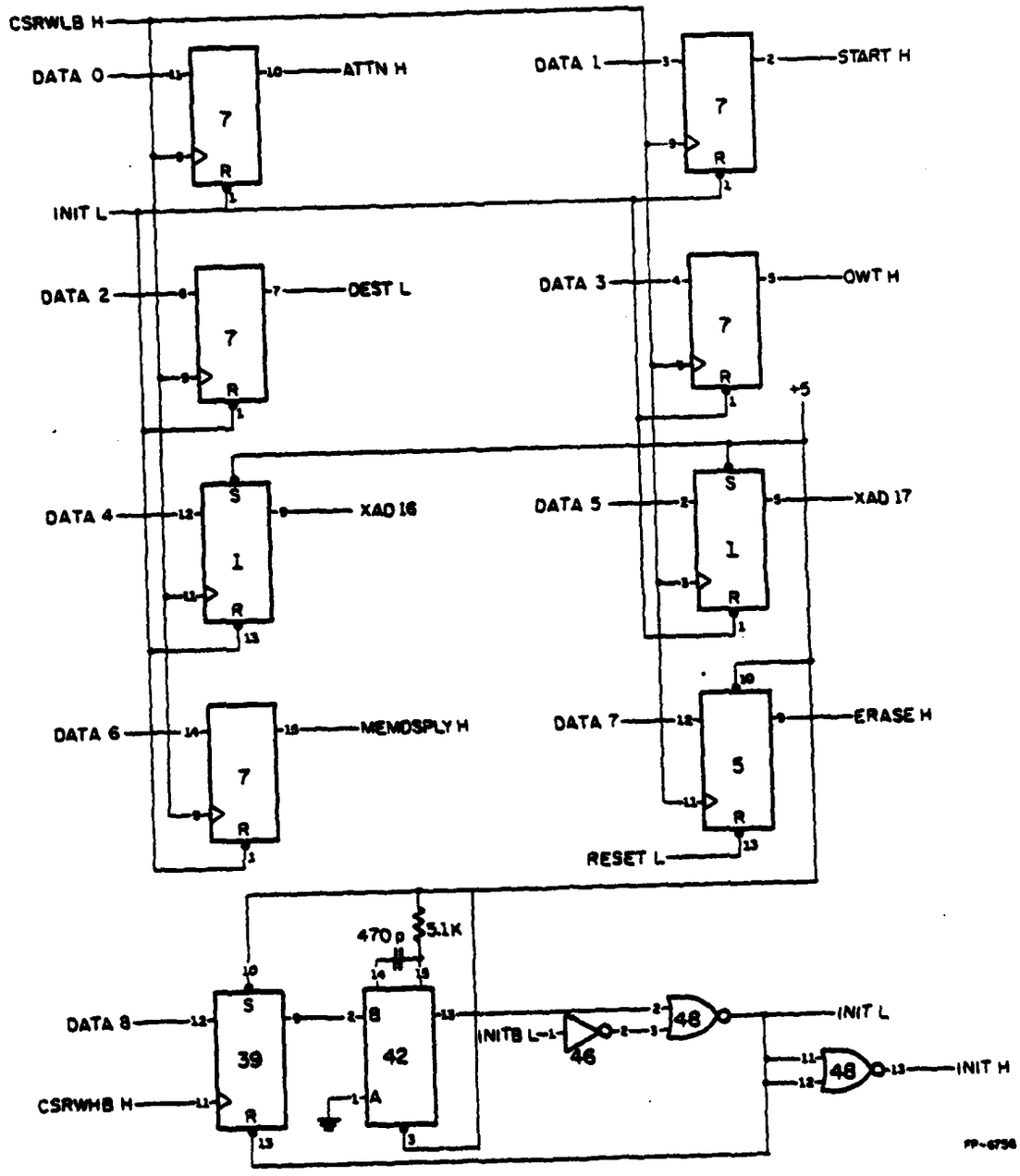


Figure 16. DMA and program control schematics.

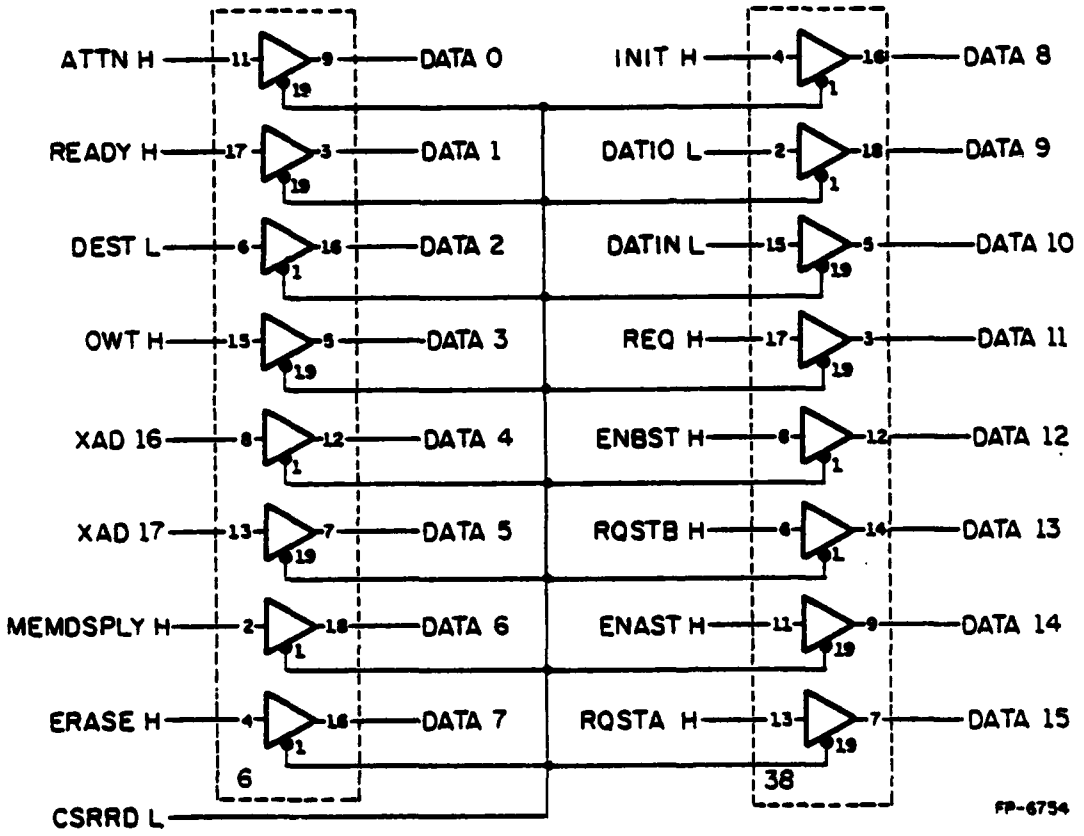


Figure 17. DMA and program control schematics.

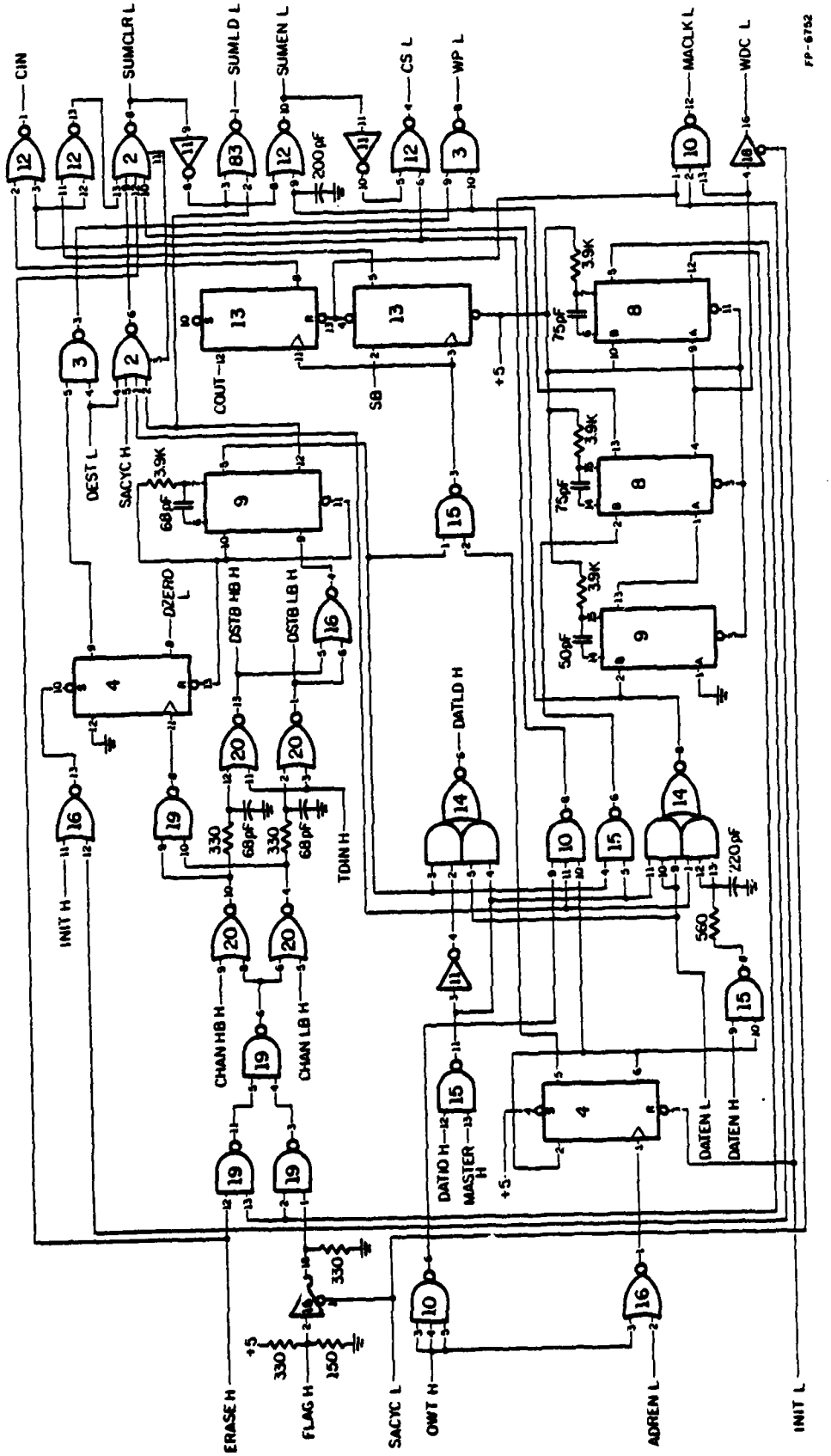


Figure 18. Master controller schematic.

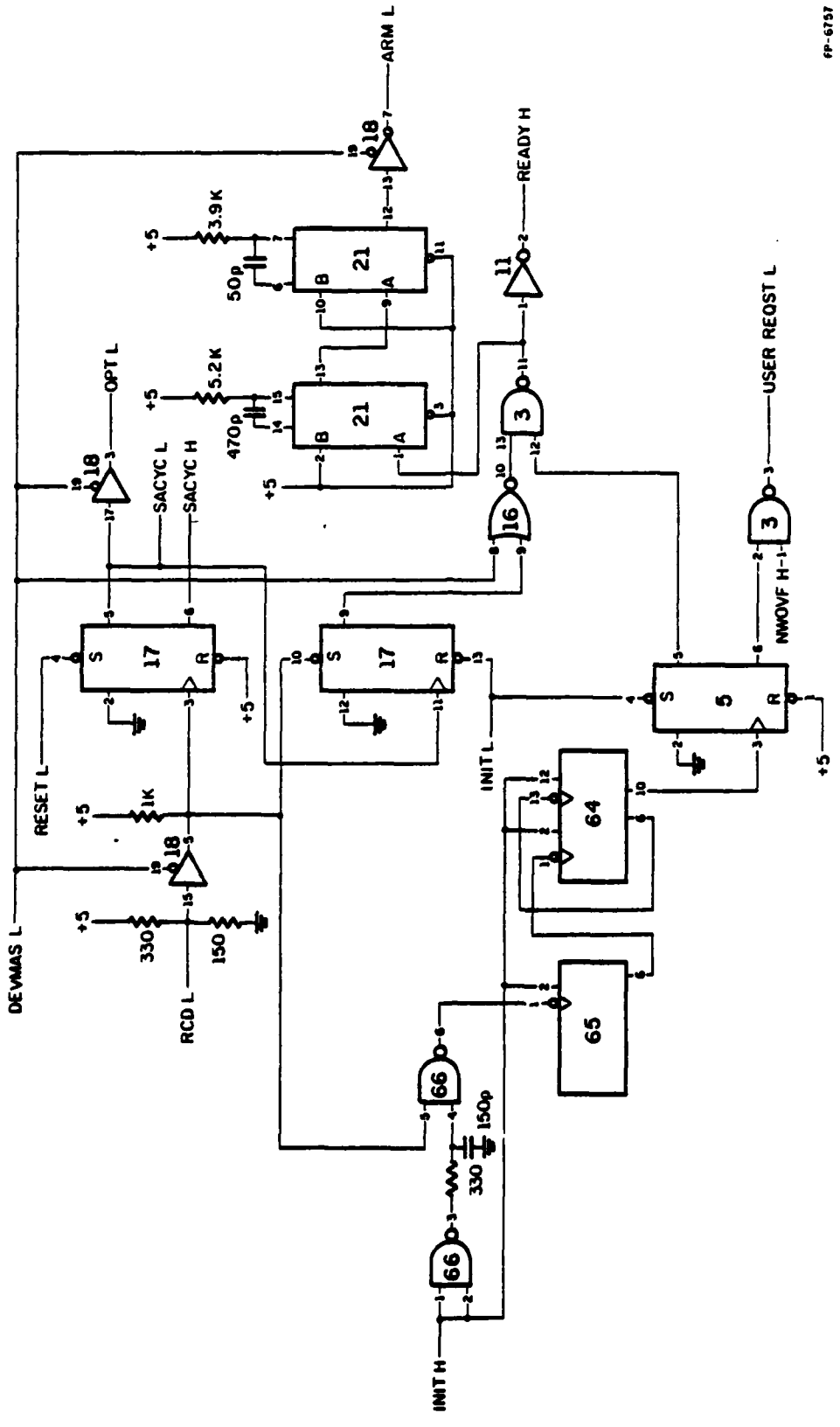


Figure 19. Master controller schematic.

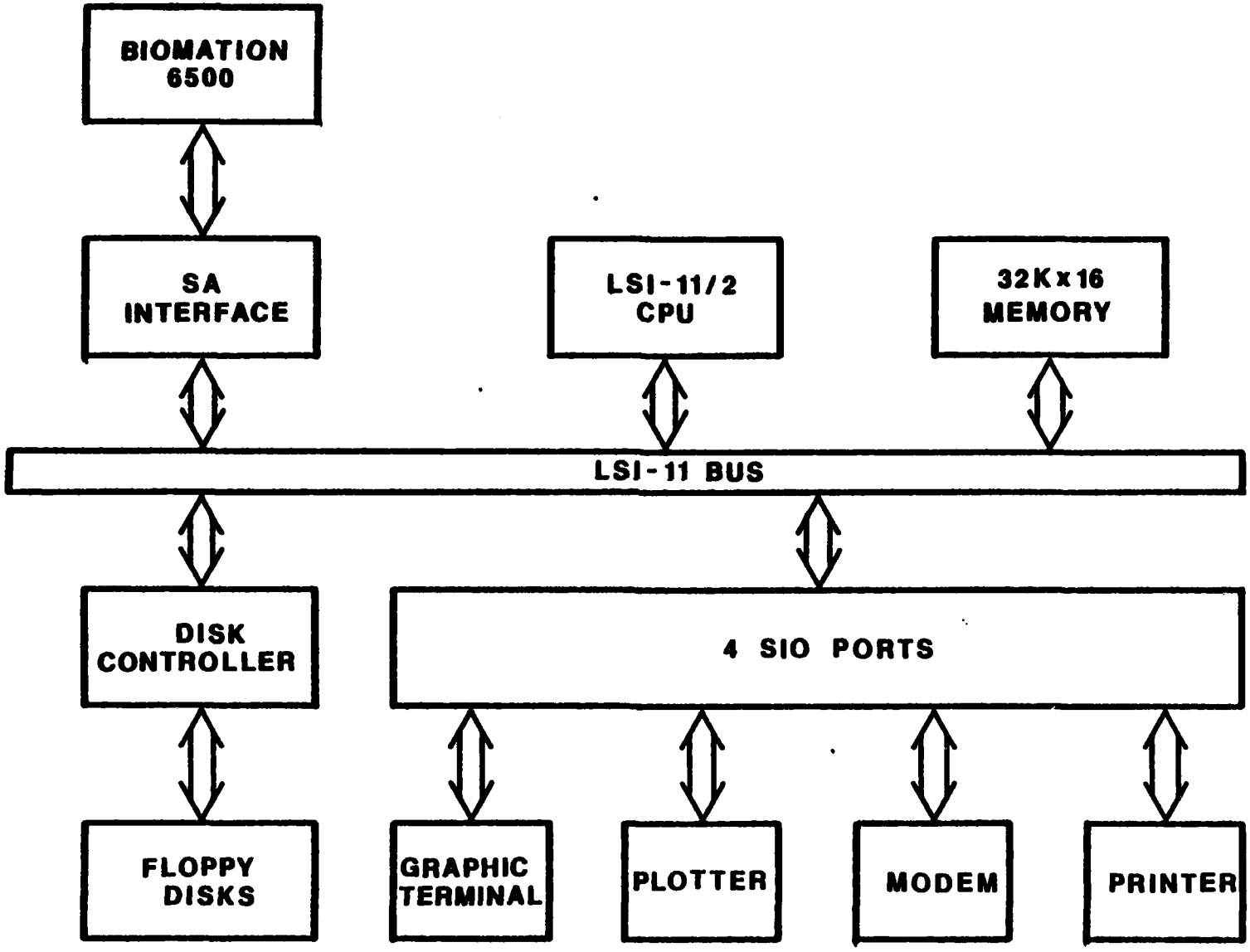


Figure 20. Total system configuration.

IC PARTS LIST

Number	Type	Number	Type
1	74LS74	31	74LS27
2	7425	32	74LS02
3	74LS00	33	10 pole DIP switch
4	74LS74	34	DC005
5	74LS74	35	DC005
6	74LS244	36	DC006
7	74LS174	37	DC006
8	74LS221	38	74LS244
9	74LS221	39	74LS74
10	74LS10	40	74LS74
11	74LS04	41	74LS74
12	74LS02	42	74LS221
13	74LS74	43	74LS221
14	74LS51	44	74LS00
15	74LS00	45	74LS02
16	74LS02	46	74LS04
17	74LS74	47	74LS132
18	74LS244	48	74LS28
19	74LS00	49	74LS86
20	74LS02	50	74LS273
21	74LS221	51	74S412
22	DC005	52	74S412
23	DC005	53	74LS273
24	5 pole DIP switch	54	74LS244
25	74LS240	55	74LS244
26	DC003	56	74LS374
27	DC010	57	74LS283
28	74LS240	58	74LS283
29	DC004	59	74LS283
30	74LS27	60	74LS283

Number	Type
61	74LS283
62	74LS244
63	74LS244
64	74LS393
65	74LS393
66	74LS132
67	74LS393
68	74LS374
69	2114
70	2114
71	2114
72	2114
73	74LS374
74	AD561
75	74LS02
76	4 pole DIP switch
77	AD561
78	AD509
79	74LS374
80	AD509
81	7912
82	LM311
83	74LS02

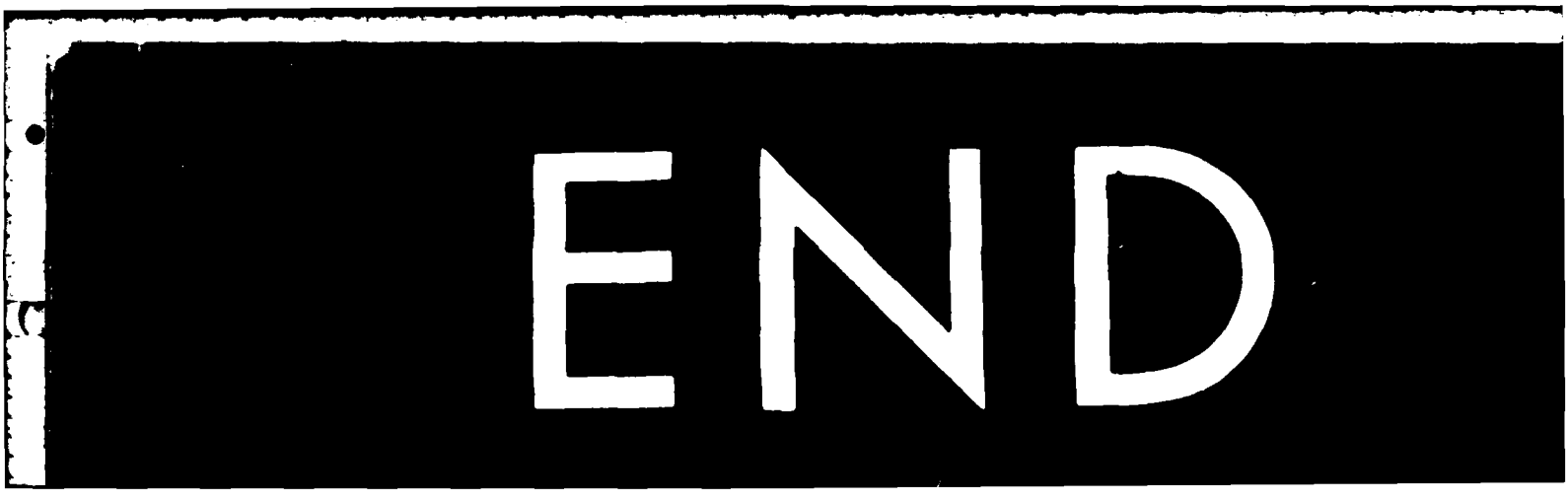
All resistors are ¼ w 5% carbon

All capacitors < 1000 pf mica 5%

.001 µf ceramic disc

.01 - 1 µf monolythic 50V

> 1 µf tantalum


END